Contents

CLOUD COMPUTING (VIII-SEM)

UNIT 1

INTRODUCTION TO SERVICE ORIENTED ARCHITECTURE, WEB SERVICES, BASIC WEB SERVICES ARCHITECTURE, INTRODUCTION TO SOAP, WSDL AND UDDI, RESTFUL SERVICES – DEFINITION, CHARACTERISTICS, COMPONENTS, TYPES

Q.1. Define SOA. What are its benefits ?

Ans. A service-oriented architecture (SOA) service exposes a clearly defined activity like credit card validation to consuming business applications that might need to perform that function (such as an order processing application). At the core of the service-oriented architecture philosophy is the modularization of business functions for greater flexibility, manageability, and reusability. With thoughtful engineering and an enterprise point of view, SOA offers positive benefits which are as follow –

(i) Language-neutral Integration – The foundational contemporary Web services standards use extensible markup language, which is focused on the creation and consumption of delimited text. Regardless of the development language used, these systems can offer and invoke services through a common mechanism. Programming language neutrality is a key differentiator from past integration approaches.

(ii) Component Reuse – Given current Web service technology, once an organization has built a software component and offered it as a service, the rest of the organization can then utilize that service. With proper service governance, emphasizing topics such as service provider trust, service security, and reliability, Web services offer the potential for aiding the more effective management of an enterprise portfolio, allowing a capability to be built well once and then shared. Multiple components can be combined to offer greater capabilities in what is often termed "orchestration".

(iii) Organizational Agility – SOA defines building blocks of software capability in terms of offered services that meet some portion of the

organization's requirements. These building blocks, once defined and reliably operated, can be recombined and integrated rapidly.

(iv) Leveraging Existing Systems – One common use of SOA is to define elements or functions of existing application systems and make them available to the enterprise in a standard agreed-upon way, leveraging the substantial investment already made in existing applications. The most compelling business case for SOA is often made regarding leveraging this legacy investment, enabling integration between new and old systems components.

Q.2. What do you understand by service oriented architecture (SOA) ? How it support cloud computing ? Explain. ***[R.G.P.V., Dec. 2013 (BE)]***

Ans. SOA is a specification and a methodology for providing platform and language-independent services for use in distributed applications. SOA describes a message-passing taxonomy for a component-based architecture that provides services to clients upon demand. Clients access a component that complies with SOA by passing a message containing metadata to be acted upon in a standard format. The component acts on that message and returns a response that the client then uses for its own purpose. Service consumers and service providers do not pass message directly to each other. Implementations of SOA employ middleware software to play the role of transaction manager and translator. This architecture does not contain executable links that require access to a specific API. The message presents data to the service and the service responds. It is up to the client to determine if the service returned an appropriate result. A SOA is then seen as a method for creating an integrated process as a set of linked services. The component exposes itself as an "endpoint" to the client.

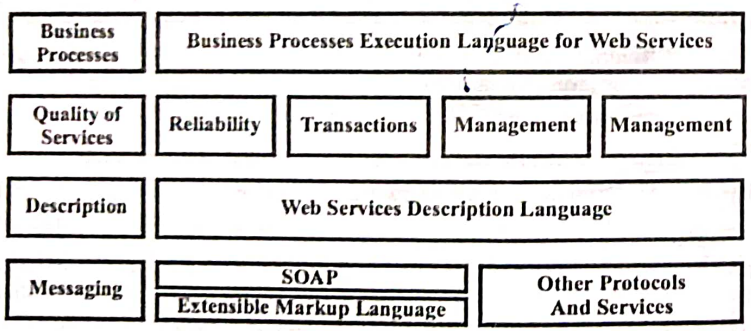


Fig. 1.1

A protocol stack for an SOA architecture and how those different protocols execute the functions required in the SOA is shown in fig. 1.1. In the figure, the box labeled other services cloud include Remote procedure calls, Jini,

common object request broken architecture, distributed common object model, windows communication foundation, Representational state transfer, and other technologies and protocols. It is this flexibility and neutrality that makes SOA so useful in designing complex applications.

Q.3. What are the basic and key features of SOA ?

Ans. **Basic Features** – The basic features of SOA are as follows –

(i) Separate functional entity.

(ii) Access to large amounts of data under a low frequency.

(iii) The way to transmit message based on text.

Key Features – The key features of SOA are as follows –

(i) Providing application development and integrated architecture, strategy on the level of thought way, and no longer only focusing on the technical level.

(ii) Providing a model of component that is functional unit what can be used.

(iii) Providing the way to integrate the functional units and make them interact with each other.

(iv) Providing the standardized interfaces, which are independent of the operating systems, hardware platforms and programming languages.

Q.4. Describe the integration between SOA and cloud computing.

Ans. The core of SOA is the service, which packages all components as services. It solves the problem of business requirement in the form of service. And the specifications of Web service which is as one of the realizable mechanisms of SOA, have many characteristics, such as standard, simple, cross-platform, cross-operating system and cross-language. It is generally followed by the grid computing, cloud Computing. Thus, cloud computing as a new Web service based on the SOA, can easily coexist with other Web services, and can interoperate with each other. But for users, they just need to know what kind of service they need, just need to understand how to access services without having to know who is to provide services in the end.

SOA is a software-oriented structure, allowing enterprises to get more than one service (a service that is a function module) from the service provider, and then forming a service that they need through a combination of different mechanisms. All resources are services in the cloud computing – the enterprises can purchase hardware service, platform services, and software services from the cloud computing provider, and make the resources they have bought as a service provider. The cloud services offer a number of services which are available to be integrated for the SOA. Since the function provided by a single cloud service is often unable to meet the business needs, so we need to integrate

and combine cloud services to solve the specific business problems, and SOA services can be effectively integrated and combined. So it is both necessary and possible to integrate the two.

It can be said that the cloud computing integrates lots of advantage of the SOA's technologies and ideas. There is the possibility of integration between cloud computing and SOA. We can see form the fig. 1.2 that there are two ways possible to integrate the SOA and cloud computing – one is SOA provide services totally depending on using cloud computing, the other one is SOA provide services partly depending on using cloud computing. It should be noted that, standards should be made between SOA and Iaas, SaaS and PaaS. Only in this way it can be possible to achieve interoperability between the two, and to achieve the blend between the two.

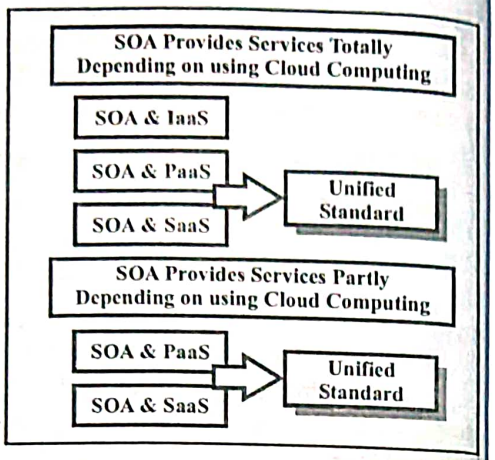


Fig. 1.2 The Possible Ways of Integration between SOA and Cloud Computing

The ideal way of integration between SOA and cloud computing is seamlessly integrating the SOA into the cloud computing platform and adopting cloud computing to provide services.

Q.5. Explain the design of cloud computing framework model based on the SOA.

Ans. The emergence of SOA has effectively solved the problem of information isolated in land and legacy systems. It has not only achieved the integration and sharing of resources, but also met the various needs of the business development and made a significant contribution to the reform of the enterprise business systems. However, with the continuous development of the computer and Internet technology, with the increasing amount of resources, the traditional approach has been unable to meet the demand. The emergence of cloud computing offers the possibility to solve this problem, the cloud computing architecture based on SOA cannot only make full use of hardware and software resources existed, but also save costs, be good for maintenance, which reflects its practical significance. We design the cloud computing architecture based on SOA, the architecture includes multiple levels containing the applications and hardware systems. An alternative reference architecture shown in fig. 1.3, bottom-up can be divided into hardware platform layer, cloud management layer, SOA structure layer and application service layer.

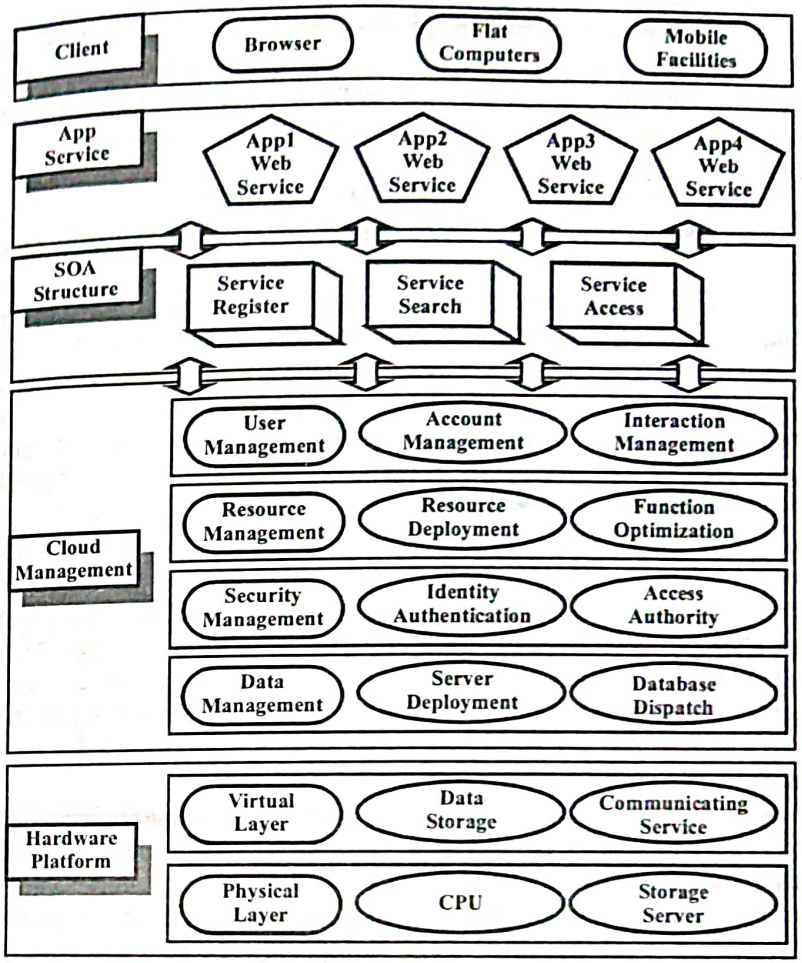


Fig. 1.3

(i) Hardware Platform Layer – That is, the so-called IaaS in the cloud computing, it includes physical and virtual service layer. It is the basic layer in the structure of cloud computing. Its main role is providing clients with virtualized storage resources, computing resources, network resources, and communication resources. And the hardware it provides containing storage servers, CPU, computers, and so on. The CPU resources can provide the services of data computing and processing for the users, the capacity and speed of data processing is much better than the client's computer. The memory resources can provide for users far more storage capacities than the client's computer cluster can provide distributed computing, greatly improving the speed of computing.

(ii) Cloud Management Layer – That is, the so-called PaaS in the cloud computing. The function of this layer is that, it has the capability of user management which contains account management, user interaction management; cloud resource management which contains resource deployment, performance optimization; cloud security management which contains identity authentication, access authority management, and cloud data management which contains server deployment, database dispatcher management. This layer ensures that the various resources can be applied safely and efficiently.

(iii) SOA Structure Layer – This layer can package the cloud resource management, cloud security management, cloud data management, User management which come from the cloud management layer into the standard Web services, then put them into SOA system for the efficient use and management. This layer includes the components of service registration, service search, service access, and so on, enabling the system to quickly respond to various levels of service (IaaS, PaaS, SaaS).

(iv) Application Service Layer – That is, the so-called SaaS in the cloud. This layer provides for users services which are packaged by the SOA structure layer. The end-consumers can customize or get the services they need through a specific entrance into the center of cloud computing. Through the cloud computing application service layer, end-customers can save a lot of overhead used in the application development and improving machine performance to use the much better machine performance in the area of complex computing or storage work.

In this model, we can learn to use the idea of SOA to get the cloud computing service, the core idea is the interaction between the service requesters and the service providers. The main roles include cloud computing service requester, cloud computing service providers (covering cloud computing services and basic resources) and cloud computing service repository (in the SOA structure layer). The service requestor is an application, a software module or another service requiring a service. Service provider is a virtual service in the cloud computing environment that accepts and executes requests from users. It publishes the services and interfaces contract to the service repository so that service users can discover and access the services. It contacts the basic services and infrastructure resources to complete the demands of the service requester. A service repository is a supporter for discovery a service, it contains a repository of available services and allows user to find service interfaces that they are interested.

Q.6. What are Web services ?

Ans. A Web service is a software component that is independent of the implementation and platform. It is a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards

over an Internet protocol and it is the hottest buzz-word in computing today. Prior to Web services, interoperability and integration were major hurdles in application development. Limited integration and interoperability took place amidst numerous technologies, vendors, obstacles and formats that prevented sharing of data. Then Web services technology came along and changed all that. Web services is a technology for transmitting data over the Internet and allowing programmatic access to that data using standard Internet protocols. It is this programmatic interface that allows two applications to be integrated.

The important part of Web services is that it allows a developer to include various functionalities into a program without the need of "reinventing the wheel" and without needing to know anything about the business logic or complexity of the Web service being used. Web services today are frequently just application programming interfaces (API) or Web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosted the required services. Web services implementation typically consists of two stages – service publish and service consume. In addition, for big Web services there is a third stage – service discovery.

Web services can be classified in two classes, SOAP Web services (arbitrary Web services) and Restful Web services.

Q.7. What are the components of Web service ?

Ans. There are following three major components of Web service –

(i) SOAP – The simple object access protocol (SOAP) is based on XML. SOAP is intermediate language so that the applications written in different language can talk with each other. That means SOAP is language independent. SOAP is also platform independent. SOAP messages can be carried by a variety of network protocols; such as HTTP, SMTP, FTP, RMI/IIOP, or a proprietary messaging protocol. But the simplest way is to use HTTP. SOAP is a light-weight protocol that is used for data interchange between applications. SOAP is based on XML and XML is light-weight so SOAP is also light-weight.

(ii) UDDI – The universal description, discovery and integration (UDDI) is a XML based framework for describing, discovering and integrating Web services. UDDI is a directory of Web service interfaces described by WSDL, containing information about Web services.

(iii) WSDL – The Web service description language (WSDL) provides a model and a standard in XML format for describing the functionalities of the Web services. WSDL describes Web services interface and provide users with a point of contact where service resides, what the service can do, and how to invoke it. The description also includes details like data type definitions, the operations supported by the service, input/output message formats, network address, protocol binding.

Q.8. Explain the architecture of Web services with suitable block diagram.

Ans. In a Web service architecture a service description covers all the necessary details to grant the service interaction, including messages's format, the transport protocols and physical location. This interface hides service implementation details, allowing that the same service can be used independently of the underlying hardware or language, thus making Web services-based applications component-oriented, making those components available for reuse. The Web services architecture is shown in fig. 1.4.

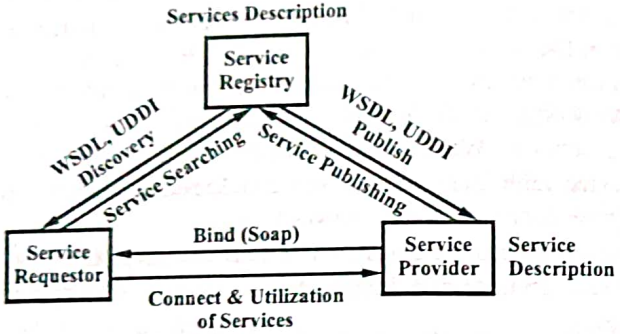


Fig. 1.4 Web Services Architecture

The three entities of Web services architecture are as follows –

(i) Service Provider – This is the service's owner from the business perspective. From the architectural approach, this is the platform that is accessed in the service request. It is also the entity that creates the Web service, being responsible to make its description in some standard format and publish its details in a central registry.

(ii) Service Requestor – It is an application that invokes or initializes some interaction with the service. It could be a web browser or even a non-user interface program such as another Web service. By using the service description it is possible to discover and invoke Web services.

(iii) Services Registry – It is the place where service providers publishes their service descriptions. Service requestors search the registries, fetching binding and description information both during the development time (static bindings) or run time (dynamic bindings).

There is the service description whose contents describes interface and implementation details, including data structures, operations and network binding informations. Also it contains data to simplify service requestor's searching process. The service is the software deployed through the network by the service provider.

Some are common operations used in Web services architecture are –

(i) Bind – When a service must be accessed, this operation invokes and initializes interaction within its caller in runtime, using binding information provided by the service description to both locate and contact it.

(ii) Publish – A service must be published in an service registry to be accessed. The service provider thus contacts the service registry to publish the service.

(iii) Discover – A service requestor finds a description of the service or queries a service registry for the required service type. A service requestor can find a service interface description in both run time or development time. Then, the necessary information regarding bindings and locales to invoke a service are found and contacted.

Q.9. Define conceptual layers in Web services.

Ans. Web services conceptual layers are shown in fig. 1.5.

(i) Service Publishing and Discovery – These two layers use the universal description discovery and integration (UDDI) standard to discovery and publish information regarding Web services.

(ii) Service Description – The description of the service is done using the Web services description language (WSDL), which defines the interface and interaction mechanisms of the service, further describing additional informations such as context, quality of service and service-to-service relationship.

(iii) XML-based Message – This layer uses the SOAP protocol as the message exchange technology standard, which stands for exchange of information in a distributed, decentralized environment.

(iv) Network Tier – It is the base layer that represents protocols such as HTTP, FTP, SMTP, POP, etc. This tier is used accordingly to the needs of the applications – security, availability, performance and reliability.

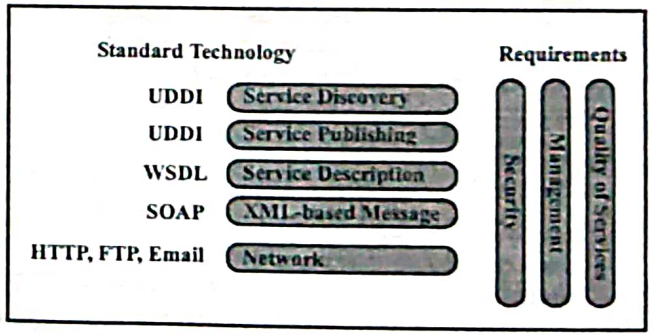


Fig. 1.5 Web Services Conceptual Layers

Q.10. What is SOAP ?

Ans. The simple object access protocol (SOAP) is a messaging protocol that allow applications to communicate using HTTP and XML. It represents a fundamentally stateless, one-way message exchange paradigm between nodes. By combining one-way exchanges with features provided by the underlying transport protocol and/or application specific information, SOAP can be used to create more complex interactions such as request/response, request/multiple response, etc.

The process of invoking Web services is very important, therefore the SOAP protocol is established to exchange messages between service providers and consumers. It is a structured XML message format for exchanging data in a distributed environment. It uses an underlying transport protocol (HTTP, SMTP etc.) through binding. There are two versions of SOAP – SOAP version 1.1 and SOAP version 1.2.

There are three main types of SOAP nodes as follows –

(i) SOAP Sender – Generates and transmits a SOAP message.

(ii) SOAP Receiver – Receives and processes the SOAP message and it also may generate SOAP response, message or fault as a result, and

(iii) SOAP Intermediary (Forwarding or Active) – It is both, a SOAP receiver and a SOAP sender. It receives and processes the SOAP header blocks targeted at it and resends the SOAP message towards an SOAP receiver.

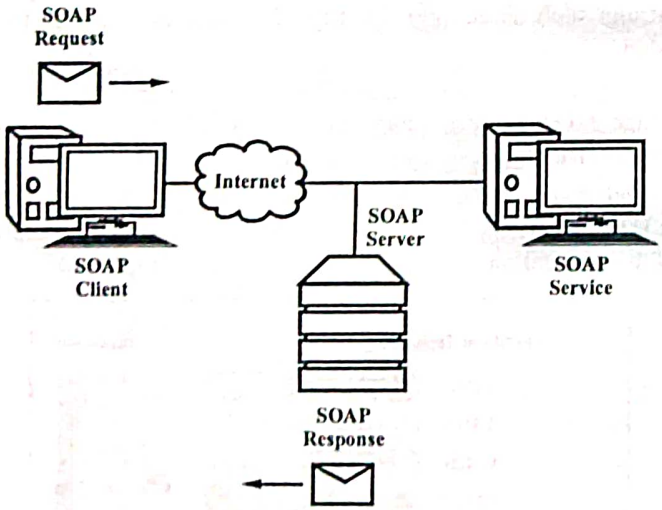


Fig. 1.6 SOAP Architecture

The SOAP message has a structure, which is characterized with two SOAP-specific sub-elements within the overall SOAP Envelope (env:Envelope), namely a SOAP Header (env:Header) and a SOAP Body (env:Body).

SOAP architecture is shown in fig. 1.6.

SOAP client is a program that creates a XML document containing the necessary information to invoke remotely a method within a distributed system (it could be a WEB or application server). ***SOAP server*** is responsible for executing a SOAP message and acts like a interpreter and distributor of documents. ***Mensagem SOAP*** is the basic communication form between SOAP nodes. SOAP requests to invoke a method on a service, including in parameters and SOAP responses from a service method, including return value and out parameters, errors from a service.

SOAP is a lightweight independent protocol. It is independent and lightweight because it does not matter what OS or what platform is the service used from – if responds in the same way in any platform or OS. All this is possible because of XML and HTTP protocols.

Q.11. What are the types of SOAP messaging request ? Explain.

Ans. There are two types of SOAP messaging requests – remote procedure call (RPC) and document request. These are as follows –

(i) Remote Procedure Call – A remote procedure call represents execution of a procedure in another remote address, usually on another computer in the same network, which is previously coded and it is called as a normal procedure local call. Thus, the programmer will only have to develop the code once, and it does not matter if the call is performed in local or remote circumstances.

This procedure represents a client-server model interaction, which is implemented through a request/response methodology. These requests and responses are formatted in XML. Usually, this communication is synchronous, which means that when a request is sent, the app is blocked until the response is processed and returned.

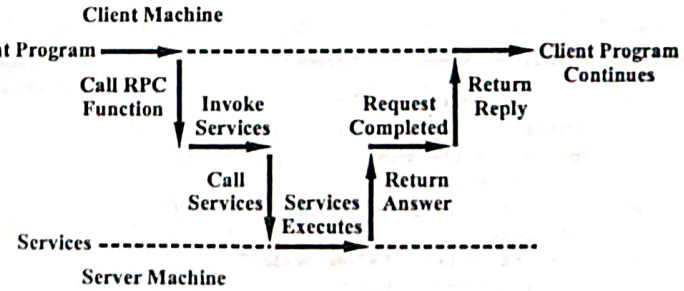


Fig. 1.7 RPC

(ii) Document Requests – While transmitting information from the client to server or vice versa through document requests, the XML document is passed in the body of the SOAP message instead of as parameter.

For example, a service named PurchaseOrder expects a document (XML document) as the input message. When the request is sent through SOAP message, requesting the PurchaseOrder operation, it must contain a purchase order document as input in the SOAP message. The request is processed as soon as it reaches the server, and when processing is done, another XML document is returned as response, which might contain any kind of information related to that purchase.

Q.12. Explain the working process of SOAP messaging.

Ans. The SOAP protocol could be potentially used in combination with a myriad of another protocols, such as HTTP, SMTP, etc. It supports remote procedure calls (RPC), and asynchronous messaging style functionality also. The SOAP data model offer data type definition to the most used structures like string, integer, float, double e date. The process of translating from data (parameters and results) to XML is often called codification.

A SOAP package consists of four parts as follows –

(i) SOAP Envelope – It defines a framework that contains the message content, who can process that message and the status of obligation to process it. It is a SOAP message structure whose syntatic elements of the message are encapsulated.

(ii) SOAP Codification – It defines serialization mechanisms that could be used to exchange instances or data types defined by an application.

(iii) Binding Framework and SOAP Transport – It defines a abstract framework to exchange SOAP envelopes between applications using a simple transport protocol.

(iv) RPC SOAP – It specifies how encapsulate remote procedure calls and responses inside the message, thus invoking remote procedures.

A SOAP message is a envelope containing optional headers and a body actually containing a message with its parameters or results. The SOAP messages are written in XML as follows –

```
<env:Envelope xmlns:env="">
  <env:Body>
    <m:getLastTradePrice
      env:encodingStyle="
http://www.Shivani.org/2001/06/soap-encoding"
      xmlns:m="
http://www.Shivani.org/2001/06/quotes"
      <symbol>DIS</symbol>
      </m:GetLastTradePrice>
    </env:Body>
    </env:Envelope>
```

Fig. 1.8 SOAP Envelope Sample Code

The SOAP envelope presents the structure as shown in fig. 1.9, where we can see the parts that composes the envelope. The SOAP block is a syntatic construction used to enclosure data that constitutes logically one only computational unity. The block is identified by an external element called URI namespace. The SOAP header is a collection of zero or more blocks, whose could be redirected to some known SOAP receptor inside the message's path. Finally, the SOAP body is a collection of zero or more blocks redirected to the last SOAP receptor.

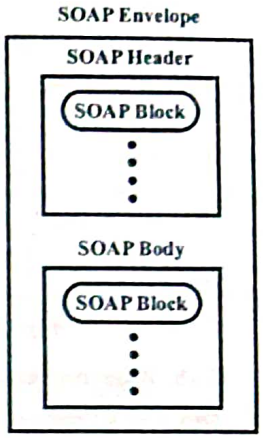


Fig. 1.9 SOAP Envelope Structure

The message navigate through the SOAP nodes as shown in fig. 1.10. Once SOAP does not guarantee routing, it knows which SOAP sender created the message and knows who will be the last receptor of the message through zero or more intermediary nodes.

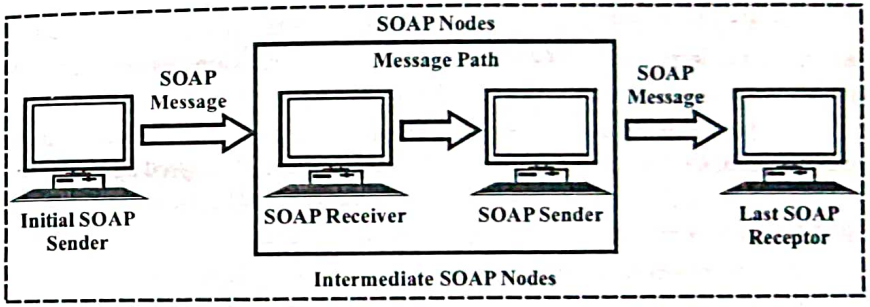


Fig. 1.10 Message's Path Through SOAP Nodes

When a SOAP node receives a message, it must execute a process, generate success or fail messages or, if necessary, return additional messages.

The process of XML message using SOAP is shown in fig. 1.11. The application requests a SOAP message and invokes the service operation through a Web service provider. The service requestor presents the message including the network address of the Web service provider. The network infrastructure ships the message to a SOAP server. This SOAP server redirects the requested message to the Web service provider. The WEB server is responsible to process a request message and then build a response. The message is redirected through the SOAP infrastructure. When a XML message arrives at the requestor, is converted to a programming language, being shipped then to the application.

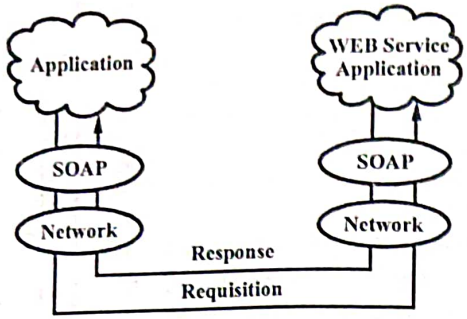


Fig. 1.11 XML Message using SOAP

Q.13. What are the advantages and disadvantages of SOAP ?

Ans. The advantages of SOAP are as follows –

(i) SOAP is platform independent.

(ii) SOAP is portable.

(iii) SOAP is capable of getting past firewalls which are totally blocking for other protocols. This is possible due to use of the HTTP protocol.

(iv) It is unlikely that future modifications of SOA infrastructure will have any impact on application using the method, as long as no significant serialization changes are made to SOAP specification.

(v) SOAP relies on open instead of vendor-specific technologies and thus enables distributed interoperability and loosely coupled applications.

(vi) SOAP is based on the open standard XML. As a consequence, SOAP becomes easily extendable and well supported.

Disadvantages of SOAP are as follows –

(i) Operation Interface – Useful information such as operation details and data are encapsulated within the services, just exposing only one endpoint of API and all operations use the POST method.

(ii) Interoperability – Since a specific service interface is defined for each service, a client must be bound to a specific WSDL. Once the WSDL has changed, the client has to follow these changes.

(iii) Complexity – It is time-consuming to serialize and deserialize native languages into SOAP messages. Furthermore, the WSDL protocol stack is also complex so that only programmers can understand how to deploy a service.

(iv) Performance – Much information in the SOAP and WSDL is redundant and meaningless. It increases the network communication volume and server side payload and it is difficult to support the proxy and cache

servers because clients cannot identify the useful information straight forwardly from the URI and HTTP.

Q.14. Explain in detail about the Web services description language (WSDL).

Ans. The Web services description language (WSDL) is a XML-based language to describe services as a collection of accessible operations through messages. It defines interfaces and service interaction mechanisms, binding protocols and network services details. It presents additional descriptions such as context, quality of service and service-to-service relationship. By using its methods, it is possible to describe an object in a transparent, implementation-independent way.

The usage of WSDL also allows a basic service description in two parts as shown in fig. 1.12.

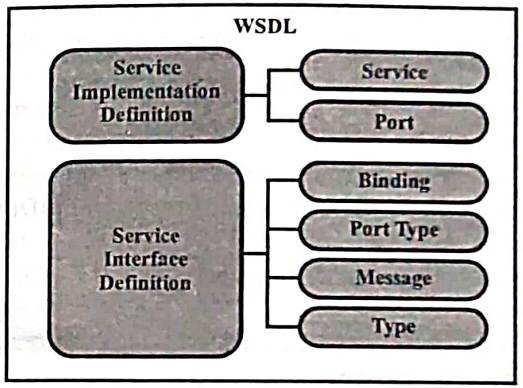


Fig. 1.12 Web Service Description Layer

This basic definition is separated in both service interface and implementation, allowing these parts to be reused separately.

(i) Service Interface Description Layer – A service interface contains a WSDL service definition. This definition allows that a interface can be used, instantiated and referenced by multiple service implementation definitions.

Binding directive describes the protocol, data format, security and other attributes to a particular service interface.

port type tag informs the Web service operation elements.

message tag is used to define data input and output regarding a operation.

type tag defines the complex data types inside a message.

A interface file describes the Web service, including the methods that are invoked, parameters that are sent and codification that is being utilized.

(ii) Service Implementation Description Layer – It is a WSDL document that describes how a service interface is implemented by a service provider. A WEB-based service is modeled as a WSDL element which contains a WSDL – ***port elements*** collection (associated port and service searching) with a WSDL – ***binding*** interface-definition element.

A implementation file describes where the Web service is installed and how it could be accessed.

Likewise the interface and implementation definitions, the WSDL specifies extensions for protocol bindings and message formats, such as SOAP HTTP GET/POST and MIME.

Q.15. What do you mean by UDDI ?

Ans. The universal description, discovery and integration (UDDI) provides a mechanism to register and locate Web service for the client and interact with them dynamically. UDDI is defined as "a set of services supporting the description and discovery of businesses, organization, and another Web service they make available, and the technical interfaces which may be used to access those services" by OASIS (The organization for the advancement of structured information standards). Two types of clients are related to UDDI registry, one who publishes the service description and another client who need the service description (using SOAP). The organizing structure of UDDI encodes three types of information about Web services – The information about the company, name and contact details are seen in a white page of UDDI, the information related to business and service types are seen in a yellow page of UDDI, and another page called Green page of UDDI provides information including technical data about the service.

Q.16. Explain in detail about UDDI with structure.

Ans. The universal description, discovery and integration (UDDI) specification is a common effort to define and create a standard service registry. This registry can be accessed by clients and these can locate all the services that they need.

The central component, called UDDI project manipulates a public, global registry called UDDI business registry. All the information maintained in the public registry are available to general queries. A private registry can add security controls to provide data integrity and prevent non-authorized access. By that way, it only can store private information, contain a subcollection from the public registry, even a combination of both. The information that is offered by business registry consists of following three components –

"white pages" containing address, contact and identifiers that are known, ***"yellow pages"*** containing industrial categorization, and ***"green pages"*** containing information.

The UDDI implementation is a Web service registry server that provides a mechanism to publish and locate services. A UDDI registry server contains categorized information about businesses, services that the lasts offers, and associations with Web services specifications. These specifications normally are also done in WSDL through a UDDI registry. The main information model used by UDDI registry is defined through a XML schema, which defines four types of information such as, business information, service information, bindings information, and service-specific information.

The information that concludes the registry of a service consists of four types of data structures. This division in a information-type fashion offers simple partitions to help in the fast search and comprehension of the different data that builds the registry. The UDDI structures are shown in fig. 1.13.

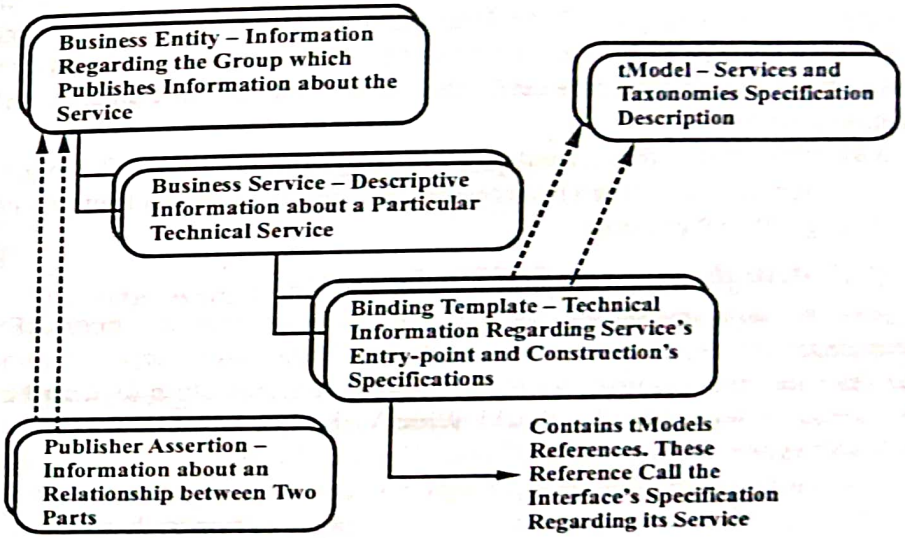


Fig. 1.13 UDDI Structure

(i) The businessEntity – This structure represents all the whole information known from an specific business or entity which publishes descriptive information about the entity, likewise its supplying services. From the XML point of view, the ***businessEntity*** is the top-level data structure whose information regards description about an business or entity.

(ii) The businessService – This represents a logical classification of the service. The element's name contains the term ***"business"*** aiming to describe the purpose of this hierarchy level in service description. Thus, each ***businessService*** structure belongs to an only ***businessEntity*** structure.

(iii) The bindingTemplate – These structures are technical descriptions of Web services that are stored via individual ***bindingTemplate***

instances. These structures provide support for one access the services remotely. The support for technology-specific parameters of the application and configuration files are encountered also. Then, it describes how the ***businessService*** uses various technical information.

(iv) The tModel – This structure is represented through metadata (data about data). The purpose of a ***tModel*** inside a UDDI registry is to provide a reference system.

(v) The publisherAssertion – Many business are not effectively represented by only one ***businessEntity***. As a consequence, many ***businessEntity*** structures can be published. Even then, they continue to represent an simple grouping and could demand that one of its relationships should be visible in their UDDI registers. In this matter, two related businesses use ***publisherAssertion*** messages, as a way to publish declarations on how they relate with each other. One or more publishers should agree that the relationship between them is valid. Both should publish exactly the same information, rendering their relationship visible.

An UDDI registry, for this matter, runs inside a UDDI server. That registry is an WEB application that can be accessed via browser or by an programmable API, through SOAP protocol.

Q.17. Write short note on RESTful Web services.

Ans. A major inspiration for RESTful Web services has been REST. This is clearly evident from the name itself. REST stands for "Representational State Transfer" and is primarily a style of software architecture for distributed hypermedia systems like the World Wide Web. Conforming to the REST constraints is termed as being RESTful. The Web is comprised of resources. A resource is any item of interest. Clients may access these resources through URLs (Uniform Resource Locators). Then a representation of these resources is returned. This representation places the client application in a state. REST is an architectural style and not standard, but uses several standards like HTTP (hypertext Transfer Protocol), URL, XML, MIME (Multipurpose Internet Mail Extensions), etc. In fact HTTP forms the very basis of REST as most of the resource handling in REST is mapped to the traditional HTTP methods like GET, POST, DELETE, etc. The latest generation of Web services has been influenced by REST to a very large extent. RESTful Web services are gaining a lot of momentum in the industry, especially with Internet companies. By using HTTP methods like PUT, GET and DELETE alongside POST, these are often better integrated with HTTP and Web browsers than SOAP based services. They do not require XML messages or WSDL service-API definitions. All of Yahoo's Web services use REST, including Flickr, del.ico.us, etc.

Q.18. Explain the architecture of RESTful Web services.

Ans. The Web application which follows the REST architecture is called as RESTful Web service. The architecture of RESTful Web services is shown in fig. 1.14. REST (Representational state transfer) as the name implies, it has to do with client and server relationship and how state is stored. REST architecture is based on the client/server architecture style. Thus, the requests and responses are built based on the transferring process of the resources. All resources are identified by unique uniform resource identifier (URI), which typically represents a document that captures the state of the resource. Generally, the REST style architecture is much lighter compared to SOAP. It does not require formats like headers to be included in the message, like it is required in SOAP architecture. In the other hand it parses JSON – a human readable language designed to allow data exchange and making it easier to parse and use by the computer. It is estimated to be at around one hundred times faster than XML. A simple JSON document is given below –

```
{
  "firstname" : "Usman",
  "lastneme" : "Saket"
}
```

There are several principles that designing RESTful Web service requires. Addressability is a REST principle where the datasets are modeled to operate as URI marked resources. Statelessness is another principle that the designer of a REST service will have to follow. This means that every transaction must be independent and must not be related to any previous transaction, as all the data required to perform and process the request are contained on that request, thus, the server will not have to maintain client session data. Uniform interface requires that an interface is uniform and standard used to access the resources, i.e. using fixed set of HTTP methods. If the service designer holds to these principles, than it is almost guaranteed that the REST application will be simple and lightweight.

RESTful Web services uses GET, PUT, POST and DELETE http methods to retrieve, create, update and delete the resources.

REST is becoming the go to for system interaction which includes the usage of RESTful Web services mostly the way cloud providers expose their services. In the present days, we can easily conclude that most of the new projects are based on RESTful

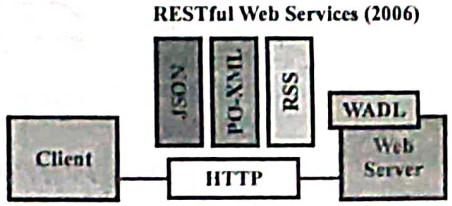


Fig. 1.14 Architecture of RESTful Web Services

architecture, in order to create and provide professional services. Large IT companies like Facebook, Google or Twitter use REST these days.

Q.19. Explain HTTP methods for RESTful Web services.

Ans. The four main HTTP methods that can be mapped to CRUD (create, read, update and delete) operations are as follows –

(i) GET – This method is used to retrieve the requested information from the server and is identified by the Request-URL. If the Requesting-URL refers to a data-processing process, then the data is returned by the requesting entity as the response, unless that text happens to be the output of the process. Use GET requests to retrieve resource representation/information only – and not to modify it in any way. It is said to be safe method because it does not cause any changes to the resources stored.

For example – HTTP GET http://www.xxyyzzz.com/users/321

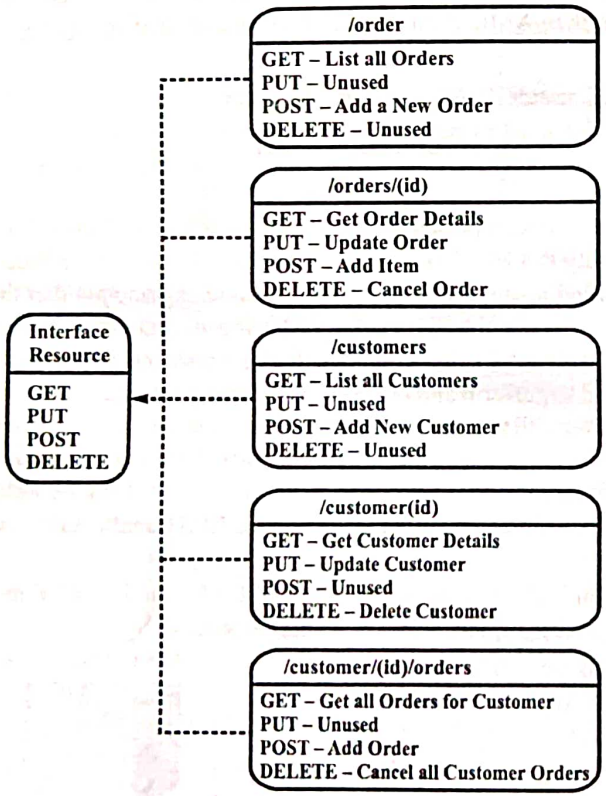


Fig. 1.15 HTTP Methods

(ii) PUT – PUT APIs is primarily used to update existing resource (if the resource does not exist, then API may decide to create a new resource

or not). If a new resource has been created by the PUT API, the origin server MUST inform the user agent via the HTTP response code 201 (created) response and if an existing resource is modified, either the 200 (OK) or 204 (No Content) response codes should be sent to indicate successful completion of the request.

For example – HTTP PUT http://www.xxyyzzz.com/users/321

(iii) POST – Use POST APIs to create new resources i.e., the data is strictly created in the server. Talking strictly in terms of REST; POST methods are used to create a new resource into the collection of resources. Ideally, if a resource has been created on the origin server, the response SHOULD be HTTP response code 201 (Created) and contain an entity which describes the status of the request and refers to the new resource, and a location header.

For example – HTTP POST http://www.xxyyzzz.com/users/321

(iv) DELETE – DELETE APIs are utilized to DELETE assets. DELETE activities are idempotent. On the off chance that you DELETE an asset, it is expelled from the assortment of assets. Over and again calling DELETE API on that asset will not change the result – nevertheless, calling DELETE on an asset a subsequent time will restore a 404 (NOT FOUND) since it was at that point evacuated.

For example – HTTP DELETE http://www.xxyyzzz.com/clients/321

Q.20. Describe key elements of RESTful Web services.

Ans. The key elements of a RESTful implementation are as follows –

(i) Resources – The first key element is the resource. Suppose a server containing the data of several employees and the URL of the Web application is http://xxx.yyy.com. So in order to access an employee record resource via REST, one can use the command http://xxx.yyy.com /shivani/101 - this command displays the data of the Shivani 101 that is stored in the Web server and fetches the data from the server in different formats such as XML or json.

(ii) Request Verbs – The request verbs describes what needs to be done with the data that is obtained from the server. The browser uses a GET request to instruct the system to get or fetch the data from the server. There are different other request verbs that can be used other than GET they are GET, PUT, POST and DELETE. So in the example http://xxx.yyy.com/shivani/101, the browser is actually using a GET verb because it wants to get the details of the employee record.

(iii) Request Headers – Headers are the different additional information that is being sent along with the data this headers also contains the

different format in which the data needs to be fetched. The headers also consists of the different authorization and authentication methods used.

(iv) Request Body – Data is posted with the request. Data is normally sent in the request when a POST request is made to the REST Web service. In a POST call, the new data that is sent is added to the server if the data is not present in the server before. Therefore, the request body contains the details of where the new data needs to be added in the Web server.

(v) Response Body – Response body is the body or the details of the response that we get from the request that is being sent. Therefore, in our example, if we were to query the Web server via the http://xxx.yyy.com/shivani/101, the web server might return an XML document with all the details of the employee in the response body.

(vi) Response Status Codes – Response status code are those codes that indicates the status of the response from the Web server. There are different codes that indicates the status of each response. For example, the code 201 indicates that the new resource has been created in the Web server without error.

Q.21. Give some advantages and disadvantages of RESTful API.

Ans. The advantages of RESTful API are as follows –

(i) REST uses smaller message format and provides cost efficiency over time and better performance because of the JSON messages which makes the communication and there is no intensive processing required.

(ii) Learning curve is reduced.

(iii) It supports stateless communication.

(iv) It is simple to learn and implement.

(v) Light bandwidth since its passes message in JSON (JavaScript Object Notation) format also it can use multiple other formats.

(vi) Efficiently uses HTTP verbs.

(vii) For security it uses HTTP standards.

(viii) REST can be consumed by any client.

(ix) It makes data available as resource.

Disadvantages of RESTful API are as follows –

(i) Compared to SOAP it does not cover all varieties of Web service standards like security, transactions etc.

(ii) REST request (especially GET) are not suitable for large amount of data.

(iii) REST is not reliable.

(iv) REST APIs end up depending on headers for state (such as to route subsequent requests to the same back-end server that handled the previous update, or for authentication). Use of headers is clumsy and ties the API to http as a transport.

(v) Latency is request processing times and bandwidth usage.

Q.22. What are the differences between SOAP and REST ?

Ans. The differences between SOAP and REST are given in table 1.1.

Table 1.1 Comparison between SOAP and REST

S.No.	*SOAP*	*REST*
(i)	Changing services in SOAP Web provisioning often means a complicated code change on the client side.	Changing services in REST Web provisioning not requires any change in client side code.
(ii)	SOAP has heavy payload as compared to REST.	REST is definitely lightweight as it is meant for lightweight data transfer over a most commonly known interface, - the URI.
(iii)	SOAP is not a wireless infrastructure friendly.	REST is a wireless infrastructure friendly.
(iv)	SOAP Web services always return XML data.	While REST Web services provide flexibility in regards to the type of data returned.
(v)	It consumes more bandwidth because a SOAP response could require more than 10 times as many bytes as compared to REST.	It consumes less bandwidth because it's response is lightweight.
(vi)	SOAP request uses POST and require a complex XML request to be created which makes response-caching difficult.	Restful APIs can be consumed using simple GET requests, intermediate proxy servers/reverse-proxies can cache their response very easily.
(vii)	SOAP uses HTTP based APIs refer to APIs that are exposed as one or more HTTP URIs and typical responses are in XML/JSON. Response schemas are custom per object.	REST on the other hand adds an element of using standardized URIs, and also giving importance to the HTTP verb used (i.e. GET/POST/PUT etc.

Q.23. Explain the types of cloud. ***[R.G.P.V., Dec. 2016 (BE)]***

Ans. Clouds can be classified into following types –

(i) Public Clouds – Cloud computing environments that are open for public use alternatively for a large industry group. Some public clouds are Google, Amazon and IBM offerings.

(ii) Private Clouds – The cloud is implemented within the private premises of an institution and uses it to provide services to the users of the institution or a subset of them.

(iii) Hybrid (Heterogeneous) Clouds – A computing environment which combines multiple clouds where those clouds keep their unique identities, but are bound together as a unit. It identifies a private cloud that has been augmented with resources or services hosted in a public cloud.

(iv) Community Clouds – These clouds are particularly intended to address the requirements of a particular industry. The cloud is characterized by a multi-administrative domain encompassing various deployment models.

Q.24. Explain public clouds in detail.

Ans. The first expression of cloud computing is formed by public clouds. Public clouds are a realization of cloud computing canonical view where the services provided are presented to anyone, at any instant and from anywhere using the Internet. They are a distributed system from a structural perspective and possibly formed by one or more data centers, on top of which the specific services provided by the cloud are implemented. Any user can easily sign-in with the cloud provider, enter details and utilize the services provided. The first class of clouds that were implemented and provided were public clouds. Public clouds provide as a feasible option for handling peak loads on the local infrastructure and provide solutions for reducing IT infrastructure costs. For small enterprises, they have become an interesting alternative. Small enterprises are able to begin their business without large upfront investments by entirely depending on public infrastructure for their IT requirements. Public clouds are used both to extend the IT infrastructure of enterprise on demand and to completely replace it.

Multi-tenancy is a basic feature of public clouds. A public cloud serves as a large number of users instead of a single one. It is required by a customer to have virtual computing environment that is separated, and possibly isolated, from the other users. This is a basic requirement to offer an effective monitoring of user activities, the other Quality of Service (QoS) attributes negotiated with users, and ensure the required performance. In public clouds, QoS management is an important feature. As a result, a considerable portion of the software infrastructure is given to monitor the cloud resources, to bill them on the basis of the contract done with the user and to keep the complete history of the cloud usage for each customer. For public clouds, these features are basic since they help providers to provide services to users with full accountability.

A public cloud can provide various type of services like software/applications, platform, and infrastructure/hardwae. As an example, SalesForce.com is a public cloud providing software as a service, Google AppEngine is a public cloud providing an application development platform as a service and Amazon EC2 is a public cloud providing infrastructure as a service. Finally, we can say that public clouds are large hardware and software infrastructure whose capability is sufficient to meet the requirements of multiple users, however, they suffer from security threats and administrative pitfalls.

Q.25. What are private clouds ?

Ans. Private clouds offer internal users with dynamic provisioning of computing resources. They are virtual distributed system that depend on a private infrastructure. Private clouds use other schemes in place of a pay-as-you-go model. They keep in-house the core business operations by depending on the existing IT infrastructure and decreasing the burden of maintaining it once the cloud has been created. Here, sensitive information does not flow out of the private infrastructure, therefore, security concerns are less critical. As the private cloud can offer services to various types of users, existing IT resources can be better utilized. The possibility of testing applications and systems at a comparatively lower price is another important aspect that comes with private cloud. When it is necessary to keep the processing of information within the premises, or it is necessary to use the existing hardware and software infrastructure, private clouds are the perfect solution. The major drawback of private deployments is the inability to efficiently address peak loads and to scale on demand.

Q.26. Explain hybrid cloud with an example.
[R.G.P.V., June 2015 (MCA)]

Or

What describes a hybrid cloud ? [R.G.P.V., Dec. 2014 (MCA)]

Ans. Hybrid cloud solution can take the benefits of both public and private clouds. Hybrid clouds permit maintaining sensitive information within the premises, exploiting existing IT infrastructures, and naturally growing and shrinking by provisioning external resources and releasing them on demand. Now, security concerns are only restricted to the public portion of the cloud, that can be used to carry out operations with less stringent constraints but that are still part the system workload. A heterogeneous distributed system resulting from a private cloud is hybrid cloud. Private cloud combines extra services or resources from one or more public clouds. That's why, they are also known as heterogeneous clouds. In this scenario, dynamic provisioning is a fundamental component. Hybrid clouds deal with scalability issues by using external resources for exceeding capacity demand. For the needed amount of

time, these resources or services are temporarily leased and thereafter released. This practice is also referred to as cloud-bursting. Here, cloud-burst refers to the dynamic provisioning of resources from public clouds.

The concept of hybrid cloud is general, but in most cases it applies to IT infrastructure. Within an IaaS scenario, dynamic provisioning means the ability to obtain on demand virtual machines to enhance the capability of the resulting distributed system and releasing them. Specifically, in case of private clouds, dynamic provisioning provides more complex scheduling algorithms and policies whose aim is also to optimize the budget spent to rent public resources. The building blocks for deploying and managing hybrid clouds are infrastructure management software and PaaS solutions. A schematic diagram of hybrid clouds is shown in fig. 1.16.

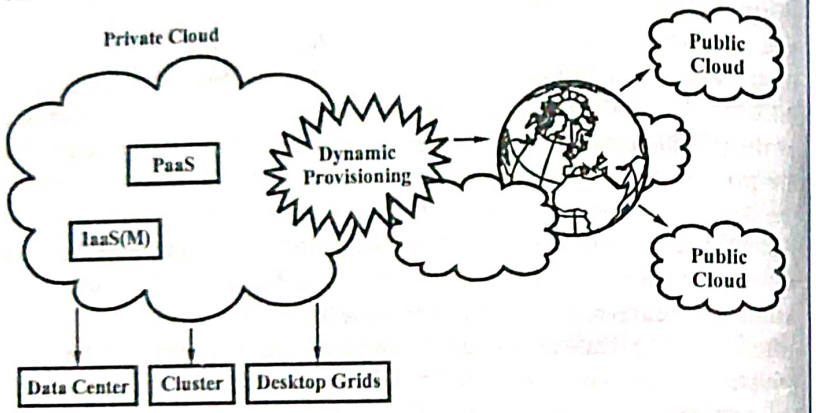


Fig. 1.16 Hybrid (Heterogeneous) Cloud

An example of hybrid cloud may consist of an organization deploying non-critical software applications in the public cloud, while keeping critical or sensitive apps in a private cloud, on the premises.

SOFTWARE AS A SERVICE, PLATFORM AS A SERVICE, ORGANIZATIONAL SCENARIOS OF CLOUDS, ADMINISTERING AND MONITORING CLOUD SERVICES, BENEFITS AND LIMITATIONS, STUDY OF A HYPERVISOR

Q.27. Describe Software-as-a-Service (SaaS) solution. How it relates to cloud computing ?

Or

What do you understand by SaaS ? ***[R.G.P.V., June 2016 (BE)]***

Ans. The concept of SaaS is precedent to cloud computing and started to circulate at the end of 90s. SaaS is a software delivery model which offers

access to applications using the Internet. It gives a way to free users from complex hardware and software management by leaving such tasks to third parties, who create applications accessible to multiple users by a Web browser. Here, customers do not install anything on their premises. Also, they do not pay considerable upfront costs to purchase the software and the required licenses. They simply access the application Website, enter their credentials and billing details, and can instantly use the application that can be further customized for their requirements. The infrastructure maintains the specific details and characteristics of each customer's application and makes available when required on the provider side.

The SaaS model is useful for applications that can be adjusted to specific needs with little further customization and serving a variety of users. This requirement characterizes SaaS as a one-to-many software delivery model. In a one-to-many software delivery model, an application is shared across several users. This is the case of Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP) applications that form general requirements for almost all the businesses. There will be similar requirements for the basic characteristics related to CRM and ERP in every enterprise, different requirements can be met with further customization. This scenario makes easy the development of software platforms offering a set of characteristics and supporting specialization and ease of integrations of new components. It constitutes the perfect candidate for hosted solutions, because the applications provided to the user are the same, and the applications itself give means to the users to shape itself on the basis of their requirements. Consequently, SaaS applications are naturally multi-tenant, which is a characteristic of SaaS. This characteristic enables providers to centralize and sustain the effort of managing large hardware infrastructures, optimizing resources by sharing the costs among the large user base, and maintaining and upgrading applications transparently to the users. Such costs constitute a minimal fraction of the usage fee paid for the software on the customer side.

The SaaS approach resides on top of the cloud computing stack. It fits into the cloud computing vision denoted by the acronym XaaS – everything as a service. Applications are provided as a service with SaaS. In the beginning, the SaaS model was useful only for lead users and early adopters. After cloud computing, there has been an increasing acceptance of SaaS as a feasible software delivery model. This results in the development of SaaS 2.0, which does not give a new technology but changes the manner in which SaaS is used.

Q.28. Write down the essential characteristics of a SaaS solution.

Ans. Like other forms of cloud computing, it is important to ensure that solutions sold as SaaS in fact comply with generally accepted definitions of

cloud computing. Some defining characteristics of SaaS include –

(i) Web access to commercial software.

(ii) Software is managed from a central location.

(iii) Software delivered in a "one to many" model.

(iv) User not required to handle software upgrades and patches.

(v) Application Programming Interfaces (APIs) allow for integration between different pieces of software.

Q.29. Discuss some applications of SaaS solutions.

Ans. The most popular SaaS applications are CRM, ERP and social networking. The most successful and popular example of CRM service is SalesForce.com, which builds on top of the Force.com platform. SalesForce.com provides either a programming language or a visual environment to organize components together for creating applications. It offers a variety of services for applications such as customer relationship and human resource management, enterprise resource planning, and several other features. Apart from the basic features, the integration with third-party applications improves the value of SalesForce.com. Customers can search, publish, and integrate new services and features into their existing applications, particularly, through AppExchange. AppExchange makes SalesForce.com absolutely extensible and customizable. Similar solutions are provided by RightNow and NetSuite.

Social networking applications like Orkut, Facebook and professional networking such as Linkedin constitute an another important class of SasS applications. Besides, the basic features of networking, SaaS applications enable incorporating and extending their capabilities by integrating third-party applications. For the hosting platform, SaaS applications can be developed as plug-ins and presented to the users to choose which applications they like to add in their profile. Consequently, the integrated applications get full access to the network of contacts and the user profile data.

Q.30. Explain Platform-as-a-Service (PaaS) solutions in detail.

Ans. In the cloud, PaaS solutions offer a development and deployment platform for executing applications. PaaS solutions form the middleware on top of which applications are made. The main functionality of the middleware is application management. PaaS implementations automate the process of deploying applications to the infrastructure, provisioning and configuring supporting technologies, configuring applications components, and managing system change on the basis of policies defined by the user. They do not expose any service for managing the underlying infrastructure and offer applications with a runtime environment. Developer's system is designed by them in terms of applications and are not related with operating systems, hardware, and other low-level services. According to the commitments done with the users,

the core middleware is responsible for managing the resources and scaling applications automatically or on request. The core middleware exposes interfaces that permit programming and deploying applications on the cloud from a user perspective. These can be in the form of programming APIs and libraries or in the form of a Web-based interface. Fig. 1.17 provides an overall view of the PaaS approach.

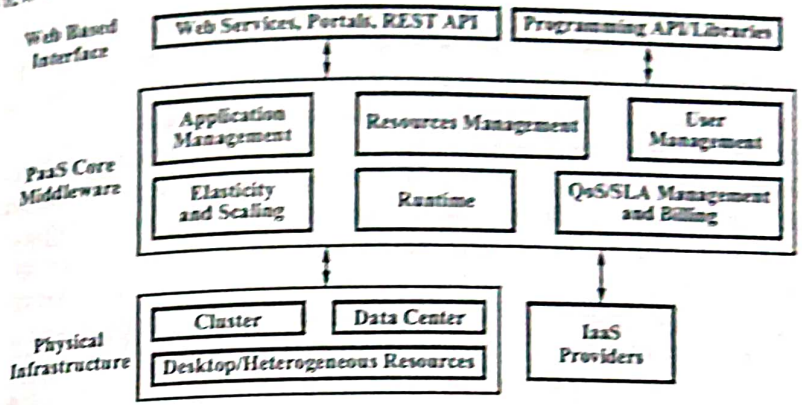


Fig. 1.17 Platform-as-a-Service

The interface exposed to the user is determined by the certain development model decided for applications. Certain implementations offer a fully Web based interface hosted in the cloud offering various services. It is possible to discover integrated developed environments on the basis of 4GL and visual programming concepts, or rapid prototyping environments in which applications are made by assembling mash-ups and user defined components, and successively customized. Other implementations of the PaaS model offer a programming language based approach and offer a complete object model for representing an application. This approach gives more opportunities and flexibility, however, generates longer development cycles. Generally, developers have the full power of programming languages with some limitations to offer better scalability and security. In this situation, the conventional development environments are used to design and develop applications, which are then deployed on the cloud by employing the APIs exposed by the PaaS provider. For better utilizing the services given by the PaaS environment, specific components can be provided together with the development libraries.

Q.31. Discuss the classification of PaaS implementations in brief.

Or

Discuss the different categories of options available in PaaS market.

Ans. PaaS solutions simply provide users with the software that is installed on the user's premises, or can provide a middleware for developing applications

together with the infrastructure. In the former case, the middleware forms the core value of the offering. In the latter case, the PaaS provider also has large data centers where applications are run. It is also possible to have vendors that provide both middleware and infrastructure and provide also only the middleware for private installations. PaaS solutions are classified into three main categories – PaaS-I, PaaS-II, and PaaS-III.

PaaS-I category recognizes PaaS implementations that entirely follow the cloud computing style for application development and deployment. They provide an integrated development environment hosted within the Web browser in which applications are designed, composed, developed and deployed. For example, Longjump and Force.com. Both of them provide the combination of a middleware and infrastructure as a platform. In PaaS-II category, all solutions that emphasize on offering a scalable infrastructure for Web applications are listed. Here, the providers APIs are used by the developers to develop applications. In PaaS-II category, the most popular product is Google AppEngine. This product offers a expandable runtime on the basis of Java and Python programming languages that have been improved with additional APIs and components to support scalability and modified for offering a secure runtime environment. An open source implementation of Google AppEngine is AppScale. Appscale offers an interface compatible middleware that has to be installed on a physical infrastructure. A similar approach to Google AppEngine is provided by Joyent Smart Platform. Engine Yard and Heroku take different approach that offer scalability support for ruby. PaaS-III category comprises all solutions that render a cloud programming platform for any sort of applications. Microsoft WindowsAzure is the most popular solution among these. It offers a framework on top of the .NET technology for building service-oriented cloud applications. In this category, other solutions are Manjrasoft Aneka, DataSynapse, Apprenda SaaSGrid, GigaSpaces DataGrid and Appistry Cloud IQ platform. All these solutions offer only a middleware with different services. Table 1.2 shows a platform-as-a-service offering classification.

Table 1.2

Category	*Vendors and Products*	*Product Type*
PaaS-I	Longjump	Middleware + Infrastructure
	Force.com	Middleware + Infrastructure
PaaS-II	Google AppEngine	Middleware + Infrastructure
	AppScale	Middleware
	Joyent Smart Platform	Middleware + Infrastructure
	Engine Yard	Middleware + Infrastructure
	Heroku	Middleware + Infrastructure
PaaS-III	Microsoft Windows Azure	Middleware + Infrastructure
	Manjrasoft Aneka	Middleware
	DataSynapse	Middleware
	Apprenda SaaSGrid	Middleware
	GigaSpaces DataGrid	Middleware
	Appistry Cloud IQ	Middleware

Q.32. Discuss the essential characteristics of a PaaS solution.

Ans. Some essential characteristics of a PaaS solution are as follows –

(i) Automation – PaaS environment automates the process of deploying applications to the infrastructure, and scaling them when required by provisioning extra resources. This process is carried out automatically and according to the SLA created between the customers and the provider. This characteristics offers ways to provision more resources.

(ii) Abstraction – The higher level of abstraction differentiates the PaaS solutions. In PaaS, the concentration is on the applications the cloud must assist. It means PaaS solutions provide an approach to deploy and manage applications on the cloud instead of a bunch of virtual machines.

(iii) Cloud Services – To help architects and developers to simplify the creation and delivery of elastic and highly available cloud applications, PaaS offerings provide them with APIs and services. Among competing PaaS solutions, these services are the main differentiators and generally entail certain components for developing applications, management, reporting, and advanced services for application monitoring. The ability to integrate third-party cloud services provided from other vendors by using service-oriented architecture is another important component for a PaaS based approach. This opportunity makes the applications development able to evolve as required by the customers and users, and able to move quickly and easily.

(iv) Runtime Framework – In PaaS solutions, this is the most intuitive characteristics and represents the software stack of the PaaS model. The end user code is executed by the runtime framework on the basis of the policies defined by the user and the provider.

Q.33. What does Infrastructure-as-a-Service (IaaS) refer to ? Explain.

Ans. Infrastructure/Hardware as a service solutions are the most popular. They provide customizable infrastructure on demand and build market segment of cloud computing. The available alternatives within the IaaS-offering umbrella are database, Web servers, load balancers and network devices from single servers to entire infrastructures. Hardware virtualization is the main technology used to provide and implement these solutions. In hardware virtualization, one or more virtual machines are suitably configured and interconnected that define

the distributed system on top of which applications are installed and deployed. Virtual machines also form the atomic components. These components are deployed and priced depending on the memory, disk storage and number of processors. The advantages of hardware virtualization are sandboxing, hardware tuning, workload partitioning and application isolation. All these advantages of hardware virtualization are provided by IaaS/HaaS solutions. IaaS/HaaS solutions decreases the administration and maintenance cost, and the capital costs from the view point of the customer. It also enables better exploitation of the IT infrastructure and offers a more secure environment for executing third-party applications from the view point of the service provider. Simultaneously, users can take benefit of the full customization provided by virtualization to deploy their infrastructure in the cloud. Virtual machines mostly come with the selected OS installed and the system can be configured with all the needed packages and applications. Apart from the basic virtual machine management capabilities, some other services can be offered. These services are workload management, SLA resource based allocation, ability to integrate third party IaaS solutions, and support for infrastructure design through advanced Web interfaces.

Q.34. Describe the basic components of an IaaS-based solution.

Ans. The basic components of an IaaS-based solution are shown in fig. 1.18. Here, the physical infrastructure, the infrastructure management software and the Web based management interface or user interface are the three principal layers.

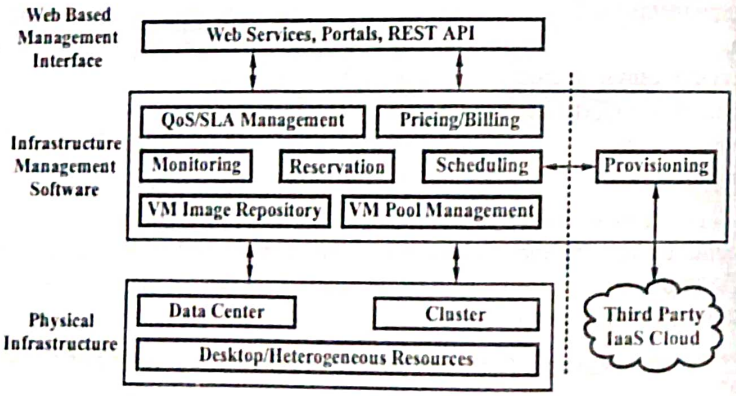


Fig. 1.18 Infrastructure-as-a-Service

The bottom layer is set up by the physical infrastructure. The management layer works on top of the physical infrastructure. The infrastructure can be of various types and the use of particular infrastructure relies on the specific use of the cloud. A service provider will probably use a large data center that has several number of nodes. A cloud infrastructure will depend on a cluster

whether it is developed in house, in a small or medium organization or within a large department. It is also possible at the bottom of the scale to consider a heterogeneous environment where various types of resources such as PCs, clusters and workstations can be aggregated. The physical layer also incorporates the virtual resources that are rented from external IaaS providers from an architectural point of view.

In the infrastructure management software layer, the main features of an IaaS solution are implemented. The most prominent function carried out by this layer is the management of the virtual machines. The scheduler plays a central role, who is responsible for allocating the execution of virtual machine instances. The scheduler interacts with the other components to carry out several other tasks. A QoS/SLA management component will maintain a repository of all the Service Level Agreements (SLAs) done with the users and together with the monitoring component is used to make sure that a given virtual machine instance is executed with the required Quality of Service (QoS). The pricing/billing component takes care of the cost of executing each virtual machine instance and stores data that will be used to charge the user. The monitoring component records the execution of each virtual machine instance and stores data needed for analyzing and reporting the system performance of the system. The reservation component records the detail of all the virtual machine instances that have been executed or that will be executed in the future. The VM repository component offers a catalog of virtual machine images that are used by the users to produce virtual instances. A VM pool manager component maintains track of all the live instances. A provisioning component interacts with the scheduler in order to offer a virtual machine instance that is external to the local physical infrastructure, if the system supports the integration of additional resources related to a third party IaaS provider.

At the top layer, the user interface offers access to the services exposed by the management layer. Generally, this type of interface depends on Web 2.0 technologies, which enable either applications or final users to access the services exposed by the underlying infrastructure. Web 2.0 applications permit developing full-featured management consoles fully hosted in a Web page or a browser. Without the human intervention, RESTful APIs and Web services enable program to interact with the service. Hence, offering complete integration within a software system.

Q.35. Explain the uses of –

(i) Infrastructure as a service

(ii) Platform as a service

(iii) Software as a service.

[R.G.P.V., Nov. 2018 (MCA)]

Ans. (i) Infrastructure as a Service (IaaS) – The infrastructure as a service layer was based on virtualization technology and provides the basic computing infrastructure of servers, processing, storage, networks and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications and possibly limited control of select networking components.

The main purpose of IaaS is to avoid purchasing, housing and managing the basic hardware and software infrastructure components, and instead obtain those resources as virtualized objects controllable via a service interface (which reduces hardware costs).

Customers are allocated computing resources in order to run virtual machines consisting of operating systems and applications.

(ii) Platform as a Service (PaaS) – Platform as a service adds a higher level to the cloud infrastructure by providing a platform upon which applications can be written or deployed. These service providers offer, in this layer, application programming interfaces (APIs) that enable developers to exploit functionality over the Internet rather than delivering complete applications. It delivers development environments to programmers, analysts and software engineers as a service. PaaS gives end users control over application design, but does not give them control over the physical infrastructure. In this services, customers may interact with the software to enter and receive data, perform actions, get results and to the degree that the vendor allows it customize the platform involved.

The main purpose of PaaS is to reduce the cost and complexity of buying, housing and managing the underlying hardware and software components of the platform, including any needed program and database development tools. PaaS gives end users control over application design, but does not give them control over the physical infrastructure.

(iii) Software as a Service (SaaS) – Software as a service provides the consumer with typical software applications that run over the cloud computing infrastructure. SaaS delivers applications through a web browser to thousands of customers rather than installed on their computer. The end user does not exercise any control over the design of the application, servers, networking, and storage infrastructure.

The main purpose of SaaS is to reduce the total cost of hardware and software development, maintenance and operations. All these services offer scalability and multitenancy. In addition, they are self-provisioning and can be deployed through public cloud deployment modules.

Examples of these three modes of cloud computing are given in table 1.3.

Examples – Amazon's elastic compute cloud (EC2) is a prominent example for an IaaS offer. It offers the user a virtual server, with the CPU, memory, storage, operating system and hypervisor or system monitoring software included.

(a) Google App Engine – Is an example for a Web platform as a service (PaaS) which enables to deploy and dynamically scale Python and Java based Web applications.

(b) Google Apps – Provides Web-based office tools such as e-mail, calendar, and document management.

(c) Rackspace Cloud – Is a cloud IaaS. It provides users with access to dynamically scalable computing and storage resources, as well as third-party cloud applications and tools.

(d) Salesforce.com – Is a cloud SaaS. It provides a full customer relationship management (CRM) application.

(e) Zoho.com – Is a cloud SaaS. It provides a large suite of Web-based applications and it is used often by organizations.

Table 1.3

Attributes	*SaaS*	*PaaS*	*IaaS*
Service providers	Google apps, Office live, G-mail, Facebook	Azure, Netsuite, Amazon web services	IBM, Amazon EC2, Xen SalesForce.com
Runtime management	By the customers	By the vendor	By the vendor
Data management	By the customers	By the developer	By the vendor
Application management	By the customers	By the developer	By the vendor
Used by	Business users	Developers and deployers	System manager
Visibility	End users	Application developers	Network architects
Type of services	Dynamic infra-structure service	Integration as a service	Dynamic application services
No of providers	Large numbers of application in the cloud	Few cloud platforms	Elite group of providers
Server management	By the vendor	By the vendor	Small

Q.36. List the advantages and disadvantages of the services on the clou
[R.G.P.V., Nov. 2018 (MCA

Ans. The advantages and disadvantages of the services on the cloud a as follows –

Advantages of IaaS –

(i) Quick addition of capacity – elastic to grow and shrink as capac requirements change.

(ii) Most hardware, network and data center infrastructure co are eliminated.

(iii) Lower labor costs are network infrastructure costs, data cen costs are ongoing facility maintenance costs are omitted.

(iv) High availability Internet connections from hosting provider.

(v) Allows existing applications to be moved into the cloud with minimum of modification.

Disadvantages of IaaS –

(i) Perceived loss of control of assets and physical security.

(ii) Network outages (local connection to Internet or the Inter itself) means complete service loss.

(iii) More Internet bandwidth required from facilities.

(iv) Possible loss of logical security, concerns of information leaka from one customer to another.

(v) Compliance auditing (i.e. SBOX) may become more compl as there is no direct control of assets located within cloud.

Advantages of PaaS –

(i) Quick addition of capacity – elastic to grow and shrink as capac requirements change.

(ii) Hardware, network and data center infrastructure costs a eliminated.

(iii) Some software costs (OS and support software) are eliminate

(iv) Initial pre-configuration of core-OS and some support softwa is eliminated.

(v) High availability Internet connections from hosting provider.

Disadvantages of PaaS –

(i) Less application flexibility as applications must conform to p defined platform template.

(ii) Only certain types of applications are suited for this mod depending on specifies of platform.

(iii) Less portability of existing applications as constrained to a sin pre-defined platform configuration.

(iv) Same security/auditing/network bandwidth and dependency concerns as with IaaS.

(v) Provider "lock-in", portability of applications between providers may prove difficult.

Advantages of SaaS –

(i) Easy and quick setup, often by non-technical people and on demand.

(ii) Easy and quick tear down, on demand when need for the application subsides.

(iii) No long-term contracts means can be used for short-term bursting.

(iv) Low per-user cost.

(v) Low client requirements.

Disadvantages of SaaS –

(i) May run slower when Internet in heavily loaded.

(ii) Often "one size fits all" customization/flexibility are limited.

(iii) Provider "lock in" is very likely, difficult or impossible to switch vendors.

(iv) Security is a concern, data leakage for example.

(v) Ability to audit and traceability are concerns.

Q.37. Explain in detail about application level security in –
(i) SaaS (ii) PaaS (iii) IaaS.

[R.G.P.V., Nov. 2018 (MCA)]

Ans. Application or software security should be a critical element of a security program. Most enterprises with information security programs have yet to institute an application security program to address this realm. Designing and implementing applications aims at deployment on a cloud platform will require existing application security programs to reevaluate current practices and standards. The application security spectrum ranges from standalone single-user applications to sophisticated multiuser e-commerce applications used by many users. The level is responsible for managing –

(i) SaaS application security

(ii) PaaS application security

(iii) IaaS application security

(iv) Application-level security threats

(v) End user security

(vi) Customer-deployed application security

(vii) Public cloud security limitations.

(i) Application Security in a SaaS – SaaS vendors provide t infrastructure and applications to users on the pay-per-use model. The c per month paid to the SaaS provider is based on the modules selected, numb of user accounts, and amount of utilization of the application.

(ii) Application Security in PaaS – PaaS vendors provide t infrastructure, application building blocks, compilers, and a runtin environment to develop and host applications. These blocks could be simil to those used internally within an enterprise; however, one needs to co certain security in applications in order to cover multi-tenancy and thousan of users who have potential access to the platform. Application security h been a problem long before the arrival of PaaS. Some of the ways to prot data in a PaaS environment are as follows –

(a) Testing for Vulnerabilities – Several tools have be developed to identify application vulnerabilities. Some good ones are describ at the Open Web Application Security Project (OWASP) site (htt www.owasp.org). It lists several, battle tested tools, to protect the Web ba applications from security threats. These can be effectively used to harden t cloud applications. OWASP is a non-profit organization, dedicated to improv application security by providing tools and best practices to discover desi and implementation defects and to protect against the flaws.

(b) Tools – The cloud provider should be able to provide to to identify security issues and scan Web pages. You must continuously sc Web pages for common security issues such as XSS and SQL injections.

(c) Logs – All activity and security events must be logged a the data must be protected through encryption. The log must be regula scanned for indications of security threats.

(d) Application Keys – All API calls to the platform or servic within must require an application key. The cloud application must ha provisions to maintain and secure the key along with the other credentials.

(e) Secure Protocols – For Simple Object Access Proto (SOAP) based messages, secure protocols such as Web services secur must be used. It provides a foundation for implementing security functi such as confidentiality and integrity for Web based application. It is maintain by OASIS (Organization for the Advancement of Structured Informati Standards), an international, non-profit consortium, which is focused on op standards adoption for applications. Cloud applications must use Secure Soc Layer (SSL), whenever possible.

(iii) Application Security in an IaaS – For application within enterprise, several internal controls exist to protect the data. In a cloud, t corresponding security controls must be coded within the application. T

section describes the security aspects for application developed in an IaaS environment. In this service providers create virtual machines (VM's) with internal or external storage devices. To meet the various types of security requirements, and mitigate incessant threats, IaaS providers offer special security tools to help application developers improve security and meet compliance requirements. These tools can be used to identify and block several threats. These include the following –

(a) dWAF – It allows a set of rules to be applied to Web-based communication to accept or drop packets based on port number, source, destination IP addresses, and other parameters.

(b) Host-based Intrusion Detection Systems (HIDS) – It monitors and reports if any user or application has circumvented the IaaS host security policy.

(c) Host-based Intrusion Prevention Systems (HIPS) – It monitors each IaaS host for suspicious activities by analysing the events within the host and takes steps to stop such activity. It blocks the malicious activity by dropping the bad packets, resetting the connection, or entirely blocking traffic from the offending IP address or network to and from the IaaS host.

Q.38. Define cloud computing. ***[R.G.P.V., Dec. 2015 (MCA)]***

Ans. The term cloud computing refers to the means of providing any and all information technology from computing power to computing infrastructure, applications, business processes and personal collaboration to end users as a service when they require it.

The term cloud in cloud computing means the set of software, hardware, networks, storage, services, and interfaces that combine to provide aspects of computing as a service.

A definition given by American National Institute of Standards and Technology (NIST) is as follows –

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

A single area of concern in cloud computing is undoubtedly be privacy and security. When your data travels over and rests on systems that are no longer under your control, you have increased risk due to the interception and malfeasance of others. You cannot count on a cloud provider maintaining your privacy in the face of government actions.

Q.39. What makes cloud computing so interesting to IT stake holde and research practitioners ? Explain it by listing essential cloud computi environment and cloud service requirements. [R.G.P.V., Dec. 2014 (BE

Or

Why organizations are adopting the cloud for business and IT perspective Explain. [R.G.P.V., Dec. 2015 (MCA

Ans. Cloud computing is an emerging style of IT delivery in whi applications, data and IT resources are rapidly provisioned and provided standardized offerings to users over the Web in a flexible pricing model. Clo computing can significantly reduce IT costs and complexities. It is a way managing large numbers of highly virtualized resources such that they resemb a single large resource. There is a greater need for IT to help address busine challenges and cloud computing can help you do all of these –

(i) Reducing Risks – Ensure the right level of security and resilien across all business data and processes.

(ii) Doing More with Less – Reduce capital expenditures a operational expenses.

(iii) Breakthrough Agility – Increase ability to quickly deliver n services to capitalize on opportunities while containing costs and managing ri

(iv) Higher Quality Services – Improve quality of service on deli new services that help the business grow and reduce costs.

Cloud services today are delivered in a user-friendly manner and offered an unprecedented scale. The payment model is pay-for-what-you-use and p as-you-go, eliminating the need for long-term contract and an up front investme This presents a less disruptive business opportunity for businesses with spi or unpredictable IT demands, as they are able to easily provision massive amou of resources on a moments notice and release them back into the cloud just quickly. Following are the some reasons for adopting the cloud –

(i) Pay per use

(ii) No hardware or software to install

(iii) No long-term commitments

(iv) Massive, Web-scale abstracted infrastructure

(v) Dynamic allocation, scaling, movement of applications.

This results in business and IT aligned benefits –

(i) Lower IT barriers to launch new business services

(ii) Provide an effective and creative service delivery model

(iii) Accelerate innovation projects that can lead to new revenue

(iv) Deliver services in a less costly and higher quality busin model, while providing service access ubiquity

(v) Rapidly deploy applications over the Internet and leverage new technologies to services when, where and how your clients want them-before your competitors do.

Q.40. Explain cloud and dynamic infrastructure.

[R.G.P.V., Dec. 2014 (BE)]

Ans. Clients can access standardized IT resources through cloud computing to deploy new applications, services or computing resources fastly without reengineering their entire infrastructure, therefore making it ***dynamic.***

Cloud dynamic infrastructure depends on an architecture that contains the following initiatives –

(i) Asset Management – Provide maximum value of important business and IT assets over their life cycle with industry tailored asset management solutions.

(ii) Service Management – Across all the business and IT assets, offer visibility, control and automation to give higher value services.

(iii) Virtualization and Consolidation – Decrease operating costs, enhance responsiveness and fully use resources.

(iv) Energy Efficiency – Across the business and IT infrastructure, deals with environment, energy, and sustainability challenges and opportunities.

(v) Information Infrastructure – Support businesses obtain objectives like information compliance, availability, retention and security.

(vi) Resilience – While quickly changing and responding to risks and opportunities, handle continuous business and IT operations.

(vii) Security – Offer end-to-end industry customized governance, risk management and compliance for businesses.

Q.41. Explain briefly the cloud infrastructure self-service.

[R.G.P.V., Dec. 2017 (BE)]

Ans. In cloud infrastructure self-service, the cloud delivers the application. Users do not require any support to manage the resources. Console is provided by the cloud provider. Users uses this console to add a server instance or extra storage or a new CPU core. It helps in decrease IT support and infrastructure. Users do not require a permitted system administrator to maintain storage, mailboxes and collaboration portal because a non-technical person can also perform these tasks. Infrastructure vendors builds the templates of cloud computing on time. These templates are obtained by users from the catalogues of cloud services. Pre-defined configuration included in these templates. These pre-defined configuration issue by user for configure the cloud service. User uses these template technical information for setup ready to use clouds. User

uses these template matter to move the applications among cloud through
cloud selfservice portal. Developers may create a catalogue of cloud serv
through cloud templates. Templates also describe about how to run sim
applications on different platform like cloud based on VMware or Ama
web.

Q.42. Discuss the different barriers of cloud computing.

[R.G.P.V., Dec. 2013 (BE), June 2015 (MC

Or

Discuss the problems associated with cloud computing.

[R.G.P.V., Dec. 2016 (BE

Ans. Some major barriers identified by IT organizations to large sc
adoption of cloud services are as follows –

(i) Security – Security concerns are similar to a traditional d
center and network in a cloud environment. The overall security risks
perceived as higher for cloud services since most of the information excha
between the cloud service provider and the organization is done over the W
or a shared network, and because IT security is entirely handled by an exter
entity. Some additional factors cited as contributing to this perception
limited capabilities for monitoring access to applications hosted in the clou
belief that multi-tenant platforms are inherently less secure than single-ten
platforms, limited knowledge of the physical location of stored data and
use of virtualization as the underlying technology, where virtualization is s
as relatively new technology.

(ii) Integration and Interoperability – Identifying and migrat
appropriate applications to the cloud is made complicated by
interdependencies typically associated with business applications. Integrati
and interoperability issues include a lack of standard interface or APIs
integrating legacy applications with cloud services. This is worse if servi
from multiple vendors are involved. It also includes software dependenc
that must also reside in the cloud for performance reasons, but which may
be ready for licensing on the cloud. There are worries about how dispar
applications on multiple platforms, deployed in geographically dispe
locations, can provide the expected levels of service and can interact flowles

(iii) Governance and Regulatory Compliance – Large enterpr
are ensuring data privacy, and still trying to sort out the appropriate d
governance model for cloud services. For large organizations, quality of serv
is still cited as a major concern. Not all cloud service providers have well defi
SLAs, or SLAs that meet stricter corporate standards. Recovery times may
stated as 'as soon as possible' instead of a guaranteed number of hours. In
cloud provider's SLAs, corrective measures are often fairly minimal and do

cover the potential consequent losses to the customer's business in the event of an outage. Another issue is the inability to influence the SLA contracts. It is impractical to tailor individual SLAs for every customer they support from the cloud service provider's point of view. The risk of poor performance is perceived higher for a complex complex-delivered application. Cloud service overall performance is dependent on the performance of components outside the direct control of both the customer and the cloud service provider.

Q.43. Define cloud services.

Ans. Any Web-based application or service offered via cloud computing is called a cloud service. Cloud services can include anything from calendar and contact applications to word processing and presentations.

Q.44. What are the advantages of cloud services ?

Ans. If the user's PC crashes host application and document both remain unaffected in the cloud. An individual user can access applications and documents from any location on any PC. Because documents are hosted in the cloud, multiple users can collaborate on the same document in real time, using any available Internet connection. Documents are not machine-centric.

Q.45. What services should be provided by cloud computing vendors ?

Ans. Cloud computing vendors provide point-solution and product offerings. On the other hand, one should provide comprehensive, asset-based solutions to deploy dynamic infrastructure, which is needed for a cloud delivery model. These services are designed to provide business outputs to our clients. The following benefits should be provided by an approach to cloud computing –

(i) Services intended at specific infrastructure workloads to support accelerate standardization of services, providing productivity gains and fast client payback on their investment.

(ii) To offer visibility, control and automation in IT and business services, a service management system is embedded with cloud services.

There should be infrastructure strategy and planning services for cloud computing to support companies plan their infrastructure workloads through the suitable cloud delivery model. Cloud strategy, cloud assessment, design and development of a cloud roadmap and return on investment (ROI) assessment by workload are specific assistance. Clients should begin with a strategy and planning consulting engagement, and a pilot implementation of a key workload as well.

Q.46. Explain the different cloud services requirements.

[R.G.P.V., June 2016 (BE)]

Ans. Major cloud services requirements are as follows –

(i) Security – The resource pooling characteristics of cloud computing permits users to dynamically share physical and virtual resources. High security concerns are usually associated with virtual environments which include virtual components like hypervisor for running virtual machines and virtual environments for virtual machines communications. An organization that contain its own IT components in it premises can apply and monitor proper security policies and controls especially for identity management. As a result, it is a priority to choose a cloud service which supports suitable and adequate access control and authentication mechanisms.

(ii) Privacy – Organizations are often uncertain when it comes to managing and storing sensitive information in the cloud. Privacy is an important issue to be addressed in the direction of raising users' trust in the cloud computing paradigm. The geographic location of data stored in a cloud environment is usually not revealed to users. This lack of transparency raises high concerns especially that privacy regulations and policies differ from one country to another.

(iii) Availability – Most organizations need maximum availability levels of their computing resources provided by the cloud services. Cloud services have recently experienced failures resulting from several factors including security attacks, natural disasters and infrastructure outages. Cloud computing environments need to make serious provisions to react quickly and efficiently to such outages and ensure continuity of cloud services. Thus, the ability to efficiently and seamlessly recover from outages is an important factor to consider when comparing cloud services.

(iv) Flexibility – Cloud computing service providers should be able to offer flexible plans and services that allow an enterprise to quickly and efficiently adapt to changes in their goals, customer demands and global changes.

(v) Archiving – Archiving provides long term storage of data that is no longer momentarily needed keeping archived data imposes high storage demands and also require additional backup and restore feature for the consumers. It may also imposes some additional costs on the services. Hence, a cloud computing service with an archiving feature and easy access to archived data is among the essential requirements for many organizations.

(vi) Scalability – Cloud scalability is one of the most important factors in providing successful cloud services. It is mainly enabled by increasing the capacity and number of IT resources.

Q.47. Discuss cloud service management in brief.

Or

How cloud services are managed ? Explain in detail.

[R.G.P.V., May 2019 (MCA)]

Ans. The control, visibility and automation required for efficient cloud delivery are provided by a service management system in both public and private implementations –

(i) Make Easier User Interaction with IT –

(a) User friendly self-service interface accelerates time to value

(b) Service catalouge allows standards that drive consistent service delivery.

(ii) Allow Policies to Lower Cost with Provisioning –

(a) Reuse and release of assets are allowed by provisioning policies

(b) Service delivery is increased by automated provisioning and de-provisioning.

(iii) Increase System Administrator Productivity – Move from management silos to a service management system.

The development of cloud deployments is prompting enterprises to either look to third party cloud brokers or assemble in-house teams to manage specialized cloud service providers mainly because of the following reasons –

(i) A mechanism is required by every service-oriented approach to permit discovery and end-point resolution.

(ii) Cloud services delivered across firewalls require a third party that uses as a service broker.

(iii) Registry or repository technology offers this where service delivery is within the firewall.

In cloud computing, leveraging service brokers will possibly become a crucial success factor because cloud services multiply and expand faster as compared to the ability of cloud consumers to manage or control them. To utilize services in a reliable way, the growth of service brokerage businesses will increase the ability of cloud consumers. Cloud service providers are expected to start to partner with cloud brokerages to make sure that they can deliver the services they develop. These cloud intermediaries will support companies to select the right platform, deploy apps across different clouds and perhaps even give cloud arbitrage services that enable end users to shift between platforms to take the best pricing.

Q.48. List the companies who offer cloud computing services.

[R.G.P.V., June 2017 (MCA)]

Ans. There are several companies which offer cloud computing services –

(i) Amazon web services (ii) Microsoft Azure

(iii) IBM cloud (iv) SAP

(v) Verizon cloud (vi) Egnyte
(vii) Dropbox (viii) Navisite
(ix) Oracle cloud (x) Salesforce
(xi) Red Hat (xii) Rackspace
(xiii) VMware (xiv) Adobe
(xv) Google cloud platform.

Q.49. What is a benefit of cloud computing over traditional computing ? ***[R.G.P.V., Dec. 2014 (MCA)]***

Or

What are the benefits of adopting cloud computing ? List some of them. ***[R.G.P.V., June 2015 (MCA)]***

Or

Discuss the benefits of cloud adoption and rudiments. ***[R.G.P.V., June 2016 (BE)]***

Ans. Benefits by adopting cloud computing are as follows –

(i) Availability of Resource – One of the most general advantage is facilitated by virtualization is resource availability. Also, resource availability helps to leverage and track resource pool under the same umbrella of resource units.

(ii) Hosted Tools – By using hosted tools, the developers and testers can simply log in to the tools and use the services over the network. Due to these tools, the tester and developers need not run, install, configure or maintain tools on their systems as they can log into the tools from any machine on the network maintaining the tools.

(iii) Self-service Capability – They are capable of self-service, once somebody deploys the cloud services. Now testing teams can use the same services over the cloud and do not have to purchase the computing services and it minimizes the procurement process. Thus, they can concentrate on the testing efforts and services.

(iv) More Mobility – Peoples can access information wherever they are, rather than having to remain at their desks.

(v) Operating Efficiency – Sometimes development teams and test teams may differ in their conventions and configuration. As a result the application behaviour to be different from the desired one also the delay services. The template based approach is more transparent, with its solution stacks or hardware, configurable applications and operating system. This approach help the teams to understand the environment better.

Q.50. Discuss the advantages and disadvantages of cloud computing. Also write down the limitation of cloud computing. ***[R.G.P.V., Dec. 2017 (BE)]***

Ans. **Advantages of Cloud Computing** – Refer to Q.49.

Disadvantages of Cloud Computing – When we use an application or service in the cloud, we are using something that is not essentially as customizable as we might wish. In addition, applications deployed on-premises still have so many features compared to their cloud counterparts, although many cloud computing applications are very capable.

All cloud computing applications have inherent latency which is intrinsic in their WAN connectivity. If your application demands huge amounts of data transfer, cloud computing may not be the best model for you. However, cloud computing applications excel at large-scale processing tasks.

In addition, cloud computing is a stateless system, as is the Internet. It is essentially unidirectional in nature in order for communication to survive on a distributed system.

Another concern is privacy and security. When data is travelling over and resting on systems that are no longer under your control, there is increased risk due to the interception and malfeasance of others.

Limitations of Cloud Computing – There are several limitations of cloud computing are given below –

(i) Network Connection – Network connection should be reliable at client side. Problem in network connectivity will affect the accessing capacity of cloud. Upload and download speeds are slower as compared to that of a local server.

(ii) Control of Data Security – The client data does not secure in public cloud because client cannot control security of his data. Hacker can affect the client data by hacking or phishing attacks in cloud. Malware can easily spread in cloud because all servers are connect to each other in cloud.

(iii) Additional Costs – Cloud computing offers many cost benefits yet there are some hidden or additional costs. Client has to pay extra for data transfer and other services.

(iv) Peripherals – Printers and scanners do not work with cloud. Many devices requires software to be installed locally to work on cloud.

(v) Integration – Integrating internal applications with those on cloud are difficult and many time it does not possible.

(vi) Generic – Public cloud provides very generic and multi-tenancy services. So many organizations cannot comfortable with it. In-house cloud

implementation is very difficult and onerous on internal resources if organizati is not large.

Q.51. What do you mean by hypervisor virtual machine (HVM) ?

Or

Explain what do you understand by hypervisor management softwa and their requirements. ***[R.G.P.V., Dec. 2013 (BE)***

Or

Explain virtualization hypervisor management software.

[R.G.P.V., June 2015 (BE)

Or

Write a detailed note on hypervisor management software.

[R.G.P.V., May 2018 (MCA)

Ans. There is a companion layer of hypervisor management softwa for each hypervisor. This layer offers various functions such as create virtua machine, move virtual machine, delete virtual machine, etc. for Powe Systems, Systems Director or the Power HMC as the hypervisor managemen function controlling the PowerVM hypervisor. For each 'Hypervisor Hypervisor Management Software' pair, a unique set of APIs and GUIs i available that is used by the client IT staff and by ISVs to create managemen services or other applications. There is a freedom to make one's ow hypervisor management software with open source hypervisors like KVM and Xen.

The foundation for virtualization on server is hypervisor. It enable hardware to be partitioned into multiple logical portion and make sure isolatio among them. Ethernet transport mechanism and Ethernet switch that ar required for virtual LAN capability are supported by hypervisor. To provid support for virtual storage, hypervisor supports virtual SCSI. A globa firmware image situated outside the partition memory in the first physica memory block at physical address zero is hypervisor. As soon as system i switched on, hypervisor takes control and collect information about CPU memory, I/O and other resources. All the specified resources and othe resources that are GLOBAL to the system are contolled and owned by hypervisor. Hypervisor allows the setup of logical partitioning and corresponding partition boundaries. Hypervisor tracks resource assigned to partition and offers isolation between partitions. Hypervisor is

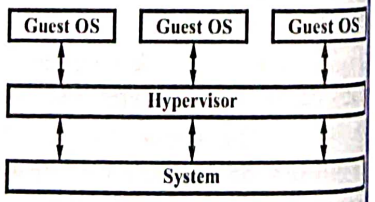


Fig. 1.19 Hypervisor Layer

used to carry out virtual memory management employing a global partition page table. The physical memory is divided into physical memory blocks and logical memory is divided into logical memory blocks. The mapping of physical memory blocks is done into logical memory blocks. The hypervisor has access to whole memory space and manages memory allocation to partitions using a global partition page table. This guarantees the isolation for memory usage.

Q.52. Describe different types of hypervisors with example and block diagram. Also enlist the advantages. ***[R.G.P.V., Dec. 2013 (BE)]***

Ans. Hypervisors are classified into two types –

(i) Type 1 (Bare Metal/Native Hypervisor) – Software systems that run directly on the host's software as a hardware control and guest operating system monitor. A guest operating system thus runs on another level above the hypervisor. This is the classic implementation of virtual machine architectures. A type-1 hypervisor is a type of client hypervisor that interacts directly with hardware that is being virtualized. It is completely independent from the operating system and boots before the operating system. Type-1 hypervisor has no host operating system because they are installed on a bare system. Examples of type-1 hypervisors are LynxSecure, Oracle VM, VirtualLogic VLX, Sun xVM Server, RTS Hypervisor, etc. Type-1 hypervisor is shown in fig. 1.20.

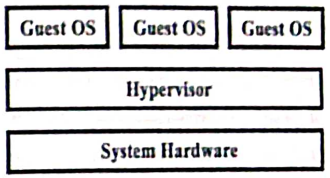


Fig. 1.20 Type-1 Hypervisor

(ii) Type 2 (Embedded/Host Hypervisors) – Software applications that run within a conventional operating system environment. Considering the hypervisor layer being a distinct software layer, guest operating systems thus run at the third level above the hardware. A type-2 hypervisor is a type of client hypervisor that sits on top of an operating system. A type-2 hypervisor relies heavily on the operating system. It cannot boot until the operating system is already up and running and, if for any reason the operating system crashes, all end-users are affected. Examples of type-2 hypervisor are Microsoft Hyper V, Wind River Simics, Containers, VMWare fusion, KVM, Xen Windows Virtual PC, Parallels Desktop for Mac, etc. Type 2 hypervisor is shown in fig. 1.21.

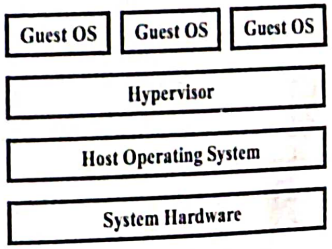


Fig. 1.21 Type-2 Hypervisor

Advantages – Following are the advantages of hypervisor technolog

(i) The hypervisor is used as a layer of abstraction to isolate virtual environment from the hardware underneath.

(ii) Hypervisor is implemented below the guest OS in the clo computing hierarchy, which means that if an attack passes the security syste in the guest OS, the hypervisor can detect it.

(iii) The hypervisor level of virtualization controls all the acce between the guests' operating systems and the shared hardware underlyi Therefore, hypervisor is able to simplify the transaction monitoring process the cloud environment.

(iv) Hypervisor controls the hardware and it is only way to acce it. This capability allows hypervisor based virtualization to have a secu infrastructure. Hypervisor can act as a firewall and will be able to preve malicious users to from compromising the hardware infrastructure.

Q.53. What virtualization platform does the chosen cloud computin service provider offer ? Does it enable the enterprise to move resource securely and freely, to and from the cloud. [R.G.P.V., Dec. 2017 (MCA)

Ans. Microsoft had developed microsoft hyper-v server in 2008. It is infrastructure/server virtualization solution. It is also known as window serve It uses a hypervisor based technique to perform hardware virtualization th leverages some methods to provide a support of different guest operatin system. Hyper-V is a component of window server 2008 R2 which instal the hypervisor as the server. Hyper-V leverages both paravirtualization an full hardware virtualization method hence hyper-V is hybrid solution compared to Xen and VMware. Hypervisor architecture is based o paravirtualized architecture. The hypervisor uses hyper calls to represe services to guest OS.

UNIT 2

UTILITY COMPUTING, ELASTIC COMPUTING, AJAX – ASYNCHRONOUS 'RICH' INTERFACES, MASHUPS – USER INTERFACE

Q.1. What is utility computing ?

Ans. A vision of computing, defining a service provisioning model for compute services in which resources like storage, compute power applications, and infrastructure are packaged and provided on a pay-per-use basis is called as utility computing. The idea of providing computing as a utility like natural gas, water, power and telephone connection has become a reality today with the advent of cloud computing.

Very similar to the traditional real world public utilities (such as electricity, water, gas and telephone), their availability and usage charges, this model of computing services and resources are available to the users whenever they need them and they will be charged based on the amount of usage. This system of packaging the storage with computational resources requires minimal or no initial hardware costs. However, the mentioned resources are rented such that they should be paid to be used. The utility computing environments are currently deployed by establishing service-oriented grids. In this case, the present resources in grids are generally clusters, where clusters are formed from the high-speed interconnection of several standalone machine.

Q.2. Write down the advantages and disadvantages of utility computing.

Ans. **Advantages** – Following are the advantages of utility computing –

(i) Low Costs – Using a utility computing provider for resources and services is far less expensive than buying and running computer hardware and operations in-house. Therefore, the users benefit reduced from operational IT-related complexities and costs.

(ii) Convenience – The client can rely on another provider for services instead of buying all the required hardware, software and licenses.

(iii) Compatibility – Companies can subscribe to a single particular service and utilize the same suite of software or service all through the entire client organization.

Disadvantages – Following are the disadvantages of utility computing –

(i) Cost – This is also considered as a disadvantage. It might be possible that a small business will choose to handle its own computing needs if the fee to access and utilize the supercomputers is high.

(ii) Educating Consumers – Inadequate advertising of utility computing to consumers, therefore, makes it difficult to sell these services to them.

(iii) Safety – There is a probability that hackers may gain access to clients' files or use services without payment.

(iv) Reliability – Utility computing companies may experience technical and equipment problems resulting in closure of business according to their financial situations. In these cases, clients may have their services curtailed.

Q.3. Write short note on elastic computing.

Ans. In cloud computing, elasticity is described as the level to which a system is capable of adapting to workload variation by offering and taking back resources the autonomic way; at every point in time the accessible resources meet the present need. It is a vital feature, which distinguishes it from earlier computing paradigms, like grid computing. This dynamic difference, so as to meet an unreliable workload, is known as elastic computing. Elastic computing is offered by cloud computing where computing resources may be scaled up and down by the cloud service supplier. Elastic computing is the capability of a cloud service supplier to provision flexible computing strength when and where required. The elasticity of such resources may be in terms of bandwidth, storage, processing power, etc.

Q.4. Describe AJAX technology in brief.

Or

Define Ajax. ***(R.G.P.V., June 2016)***

Ans. AJAX stands for Asynchronous JavaScript and XML. Asynchronous because it does not disturb the user's work, and does not refresh the full screen, JavaScript because it uses JavaScript for the actual work and XML because XML is supposed to be everywhere today. AJAX is a new technique of Web programming and its primary components are JavaScript and XML. AJAX is a technique, which describes how other technologies, JavaScript, DOM, and XML can be used together for developing interactive Web applications. These applications are based on AJAX Web application model, which uses JavaScript and XMLHttpRequest object for asynchronous data exchange.

In early days, when we used to create a Web application with JavaScript, DOM and XML technologies, the application based on these technologies were known separately. Even the end-user could not work on the Web application as desktop-based application. To overcome this, Jesse James Garrett of Adaptive Path combined JavaScript, XML, and DOM together to form a new technique, called AJAX. In this technique, the request to the Web server is sent by using the XMLHttpRequest object. This object, a part of JavaScript technology, helps in sending asynchronous request to the server. With this request, Web applications can now interact with Web server asynchronously.

AJAX can be used for making user experience better by using clever techniques for communication between a Web browser (the client) and the Web server. We can use AJAX in situations, where we want to capture the matter the user is typing or has typed, and process it while the user continues to do whatever he is doing.

Q.5. How does AJAX work ? Explain.

Or

Write short note on AJAX functionality. ***(R.G.P.V., June 2017)***

Ans. AJAX sends a request from the Web browser to the Web server whenever AJAX needs to come into the picture based on the user action (e.g., when something is typed). On the Web server, a program written in a server-side technology receives this request. Now, a program on the Web server processes this request, and sends a response back to the Web browser. Note that the user does not know that the Web browser has sent a request to the Web server while this happens. The response received from the Web server is processed by the Web browser. Then, the Web browser takes an appropriate action. This concept is shown in fig. 2.1.

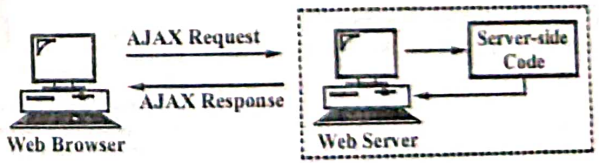


Fig. 2.1 AJAX Process

This works as follows –

(i) While the user is filling up an HTML form on the basis of some specific event, the JavaScript in the client's browser prepares and sends an AJAX request (usually called as an XMLHttpRequest) to the Web server.

(ii) While the user continues working as if nothing has happened, the Web server invokes the appropriate server-side code.

(iii) The server-side code makes an AJAX response and sends it over to the Web server.

(iv) Now, the server sends the AJAX response back to the browser while the user continues working with the remainder of the HTML form. Thereafter, the browser automatically reflects the result of the AJAX response.

Q.6. Explain the synchronous communication.

Ans. Old classic web uses synchronous communication in most of applications. Synchronous communication is good for the static web applications. Websites giving read-only information is a good example of static web. But for more interactive, dynamic and responsive web applications, synchronous communication is not suitable. Traditional web applications interactions are based on a page sequence paradigm and applications are based on a multi-page interface model.

In classic web application, generally user needs to send a request to server through some link or form, in case of synchronous communication system. In such systems, user has to wait till the server side processing. In some scenario, as a result of user query to server, a dynamic creation of contain in web page is necessary. In such cases, a full page refresh is done to update the user-desired results. In all this process, user has to suffer from very unpleasant experience of broken connection till page updating. This make-and-brake of connection continues whenever user asks for any new piece of information in synchronous communication.

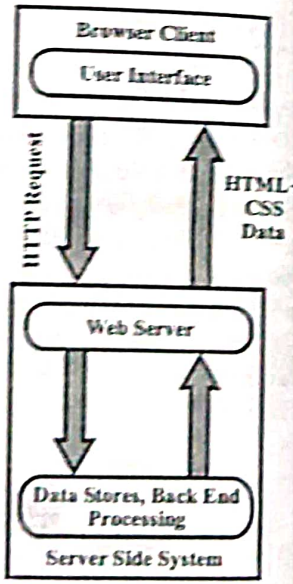


Fig. 2.2 Synchronous Communication

This pattern of request-wait-response in classic web applications is extremely troublesome. It also lowers overall productivity. The classic web application model has many drawbacks such as large response time that makes it sluggish, loss of states, usage of excessive bandwidth, less interactive nature of application and redundant data in transmission code which unnecessarily makes the code heavy.

Q.7. Discuss about the asynchronous communication used in Ajax web applications.

Ans. In asynchronous communication data can be transmitted intermittently instead of in a steady stream. The web application model in asynchronous communication is composed of independent and interactive web components, which can be created, updated, deleted and replaced independently. Whenever user requested for new information, instead of loading complete web page or redirecting user to new web page, small updates has been done without interrupting user's interaction with website. These small updates are realized through small codes of XML driving through Ajax engine. There are four scenarios by which, an effect of asynchronous communication with Ajax engine can be seen. They are as follows –

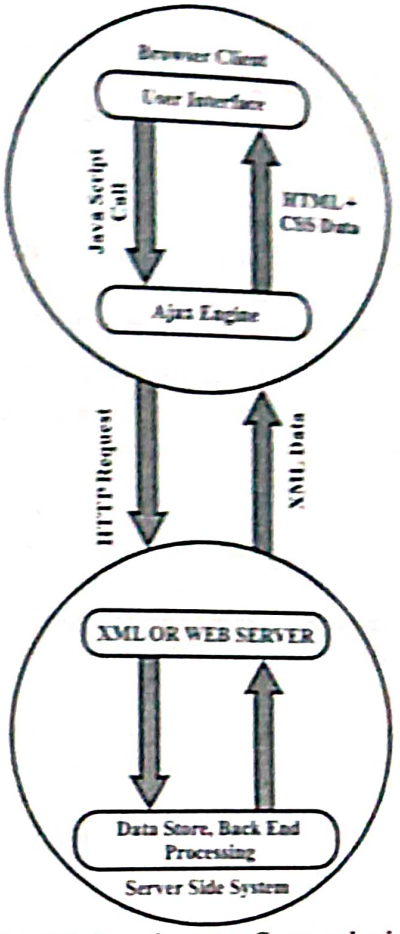


Fig. 2.3 Asynchronous Communication

(i) User Generates an Event for Ajax Engine – In this case, direct request is sent to server. And while server is responding to query, user can experience some small updates. Unlike the traditional web applications, user does not need to wait for whole page to update, instead updates can be experienced in parts.

(ii) User Generates a Request and Send Indirectly to Server – In this case also, an event is triggered for Ajax engine and after that first scenario is followed.

(iii) User Generates Event, but does not Send any Request to Server – In this case, though event is generated by user, it is not necessary that it sends request to server. In such cases JS (JavaScript) can update component with the help of HTML, CSS and JS functions, with the help of data on the same page, without triggering any event for Ajax engine.

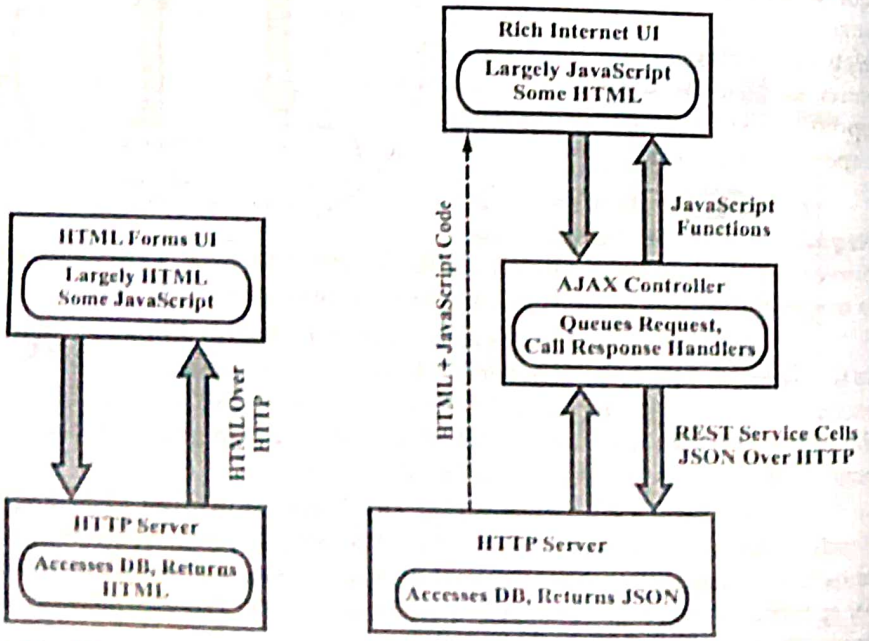
(iv) Before User Generating any Event or Request, Ajax Engine Sends a Request to Server – In this case, Ajax engine sends request to server, prior to user triggered event. Because of this process, whenever user request for specific event, results load very quickly, giving very fast and uninterrupted user experience.

Ajax engine adds asynchronous flair to the web application, by sending and receiving data from server side asynchronously. Due to this behaviour of

Ajax engine, an undisrupted and fast responding web applications can be experienced. Even if, Ajax model adds extra level in web application module, it makes overall application more responsive and interactive, instead of slowing it down. In classic web applications, whole web page needs to reloaded, on every user triggered event, Ajax eliminated this need-to-reload. Ajax also increases overall efficiency of web application, by providing updates of specific contain that user asked for, rather than updating entire page. This processes saves lots of data transmission, leading to saving of bandwidth usage. Due to this, overall process becomes much faster and user experiences more responsive and richer interaction with web.

Q.8. Explain traditional vs rich internet application.

Ans. A base HTML page is loaded along with JavaScript code that contains the remainder of the user interface. This JavaScript program renders a 'rich' user interface that can often look like a traditional client-server application. When data is required from the server, asynchronous requests are made via REST web services, which return JSON structures that are directly used by the JavaScript code running in the browser. Because of the nature of the HTTP protocol, a web server expects that an incoming request from a single client session will not be followed by another until the server has responded to



(a) Traditional Internet Application *(b) Rich Internet Application*

Fig. 2.4

the first request. If a client violates this protocol by sending many requests at a time, at best these will be ignored and at worst some poorly implemented servers may even crash! Therefore an AJAX controller is required to serialize the asynchronous requests being made to the server; each request is queued and sent to the server only after the previous request has returned with a response. Each response triggers a handler function which is registered with the controller when placing the request. Using AJAX, highly interactive yet completely browser-based user interfaces become possible. Using this approach, software as a service application can begin to provide a user experience similar to thick client applications which typically run inside the enterprise, thereby making SaaS offerings more acceptable to enterprise users. Further, using AJAX, services from multiple could providers can be integrated within the browser, using JavaScript, instead of using more complex server-side integration mechanisms based on web services. The traditional and rich internet applications are shown in fig. 2.4 (a) and 2.4 (b).

Q.9. What are the advantages and disadvantages of Ajax ?

Ans. The advantages of Ajax are as follows –

(i) Open Source – Ajax is a unique combination of existing open source technology.

(ii) Compatibility – Cross-browser and cross-platform compatibility. Also compatible with existing technologies.

(iii) Optimization – Code can be easily separately optimized in small parts.

(iv) Minimize Cost of Development – Ajax application can be developed with just simple notepad.

(v) Fast – Ajax applications are faster than traditional web applications.

(vi) Enriches user Experience – Ajax applications provide more interactive, optimized, and dynamic web contain.

The disadvantages of Ajax are as follows –

(i) JavaScript – More use of JavaScript for interaction.

(ii) Complexity – More complex than traditional web applications.

(iii) Coding – Adding extra layer to existing web model needs more coding.

(iv) Internet Explorer – Different code for internet explorer.

(v) Old Browsers – More bulky and separate code for old browsers.

Q.10. Explain the security issues of Ajax.

Ans. Ajax is combination of many technologies. Very common and generic risk in Ajax application development is not knowing all the technologies used

in Ajax building. By not knowing them, there is always risk of creating loop-holes or generating any vulnerability in the application.

The heart of Ajax is object XHR. XHR is based on JavaScript and that can cause trouble. Because JavaScript is visible in client side browser, a mal-intentional person can edit that code and can be used to validation process. One of the major problem is in server-side, if there is no validation on data. In such case, there is great risk of exposition of vulnerabilities like SQL injection. Another problem is mal-practicing of programming. Sometimes programmer writes code for client side and includes validation process for server side. This problem becomes grave danger, when web application access many different servers for information and validation code is written on the client side. In such scenario, validation code should be on the server side, or for more security, validation code should be written on one server and then from the central server, other server validation should be done, instead of user interacting directly with other servers. The validation request sent by XMR is same as in case of common web applications. Validation and analysis of request to server is very important to avoid any mishap in security.

Q.11. What do you mean by mashup ? Explain.

Ans. The term mashup is used in Web 2.0. In Web 2.0, mashup can be a web page or a web application that integrates contents such as data, presentation, or functionality from different sources to create a new service. The content can be obtained through a public interface, obtained from RSS/Atom, or provided by third party. More and more web applications have published their APIs to allow other users like software developers or web page developers to use. Developers can use these APIs to integrate data or functionalities into their own service instead of building them from scratch by themselves. The new service depends on how the developer integrates the data or functionalities from APIs. Developers can mashup with multiple data sources like maps, photos, bookmarks into a new service they desire.

Mashups are interactive web applications that take content from different sources and present them in entirely new and unique manner. The web is continuously growing and becoming more social and open. Because of this continuous growth, many websites have made their API publically available to the programmers that allow them to get information and build interactive applications. From a user perspective, mashups make it easy to consume web services. In fact, the actual service call need not even be a REST service, and may instead involve proprietary AJAX-based interaction with the service provider. In this sense, mashups make the issue of a published service standard using REST or SOAP/WSDL irrelevant; the only thing that is published is a JavaScript library which can be downloaded at runtime and executed by a client application. At the same time, the fact that mashups require downloading

and running foreign code is a valid security concern especially in the enterprise scenario. JavaScript code normally cannot access resources on the client machine apart from the browser and network, so it may appear that there is no real security threat, unlike say ActiveX controls which have essentially complete access to the desktop once a user installs them.

A good example is Google Maps, which is popular interface to many mashup applications. Google provide maps API which are used by programmer to access maps. The developer can then combine these maps with some other data source and create something new and unique. For example, if we want to create a web application to find nearest stores. A mashup can help us to use Google Maps in combination with other data source (i.e. yelp) and present the user with easy to use interface by displaying the location on the map.

Q.12. What are the types of mashups ? Explain.

Ans. There are numerous types of mashups such as business mashup, consumer mashups, and data mashups.

(i) Business Mashups – These mashups are web applications that integrate their own content, often enhanced with external web services. It is used to allow collaborative action among businesses and developers. Business mashups are secure and usually visually rich web applications that expose actionable information from diverse internal and external information sources.

(ii) Consumer Mashups – These mashups integrate different visualizations and data elements from multiple public sources to create a simple browser user interface that is more appealing consumption of information.

(iii) Data Mashup – It is opposites of consumer mashup. It integrates the same data level, whether it's integrating files, database, external web service APIs from different multiple sources into a single representation. The most common type of mashup is the consumer mashup, aimed at the general public.

SERVICES VIRTUALIZATION TECHNOLOGY – VIRTUALIZATION APPLICATIONS IN ENTERPRISES, PITFALLS OF VIRTUALIZATION

Q.13. What is virtualization ? ***[R.G.P.V., Dec. 2016 (BE)]***

Ans. The technique of masking and abstracting physical resources is known as virtualization. Virtualization makes easy the infrastructure and accommodates the increasing growth of business and technological changes. It improves the capability and utilizations of IT resources, like networks, servers, or storage devices, beyond their physical limits. Virtualization facilitates resource

management by storing and pooling resources for high utilization. It make them seem as logical resources with increased capabilities.

In case of infrastructure-based services, virtualization technology is considered as one of the fundamental components of cloud computing. Virtualization enables creation of secure, customizable, and isolated execution environment for running applications, without influencing other user's applications. In addition, virtualization technologies not only give a virtual environment for executing applications, but also for memory, storage, and networking. Virtualization offers a great opportunity to construct elastically scalable systems that are capable of provisioning additional capability with minimum costs. Thus, it is largely used to provide customizable computing environment on demand.

The three main components in a virtualized environment are guest, virtualization layer and host. The system component that interacts with the virtualization layer is represented by the guest. The virtualization layer is responsible for recreating the same or a different environment where the guest will operate. The original environment where the guest is supposed to be managed is represented by the host.

Q.14. Define virtualization ? Why virtualization is important in cloud computing ? ***[R.G.P.V., May 2019 (BE)]***

Ans. Refer to Q.13.

In cloud computing environment a very popular concept is used known as virtualization. Cloud systems use virtualization concepts as a tool to enable better functionality and more advanced features within and across latest technology.

Virtualization is critical to cloud computing because it simplifies the delivery of services by providing a platform for optimizing complex IT resources in a scalable manner, which is what makes cloud computing so cost effective.

Virtualization has three characteristics that make it ideal for cloud computing–

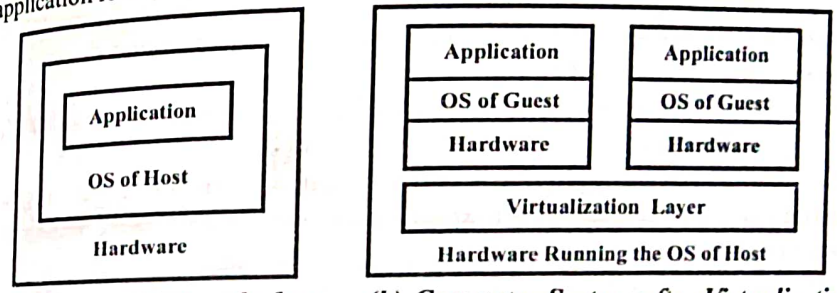
(i) Partitioning – In virtualization, we can use partitioning to support many applications and OS in a single physical system.

(ii) Isolation – Because each virtual machine is isolated, each machine is protected from crashes and viruses in the other machines.

(iii) Encapsulation – This can protect each application so that it does not interfere with other applications. Using encapsulation, a virtual machine can be represented as a single file, making it easy to identify and present to other applications.

Q.15. Discuss the architecture of a computer system before and after virtualization. ***[R.G.P.V., Dec. 2017 (MCA)]***

Ans. A traditional computer system runs with hardware, host operating system. Operating system is designed according to hardware architecture. After virtualization, operating systems can run on same hardware and manage different applications and independed from host operating system. This is possible through additional virtualization layer software. This virtualization layer is also called hypervisor or virtual machine monitor. The virtual machines are appearing in the external boxes where applications run with their own guest operating system over the virtualized hardware resources like CPU, memory and input output resources. For virtualization, the main function of software layer is to virtualize the host machine physical hardware into virtual resources to be used by the VMs exclusively. This requires more steps for implementation, the virtualization software makes the virtual machine abstraction through interposing a virtualization layer at many steps of a computer system. General virtualization layers include the instruction set architecture level, application level, hardware level, operating system level, library support level.



(a) Computer System before Virtualization

(b) Computer System after Virtualization

Fig. 2.5

Q.16. How are virtualization and cloud computing interrelated ? Explain virtualization in cluster. ***[R.G.P.V., Dec. 2017 (BE)]***

Ans. Cloud computing can be perform without virtualization. Some hardware, OS and application clusters provide cloud services but it will be very expensive and difficult. They provide limited features only but need a lot of work. Hypervisor vendors like VMware, Microsoft and Citrix systems focus on the technology of cloud, hence virtualization and cloud computing interrelate. Cloud computing is a growing technology and it will change the world of smart computing.

Virtual Cluster – A physical cluster is a group of servers connected with each other through network. Virtual cluster is a network of VMs installed at distributed servers from one or more physical clusters. In virtual cluster,

virtual machines are logically connected through a virtual network acro many physical networks. Physical machines or a virtual machine hosted multiple physical clusters create virtual cluster. Virtual cluster has distin boundaries.

There are following design issues of virtual clusters –

(i) Live migration of virtual machines.

(ii) Virtual clusters dynamic deployment.

(iii) Virtual clusters memory and file migrations.

Dynamically provisioning virtual machines to a virtual cluster has followi characteristics –

(i) Physical machines or virtual machines running different operati sysems like Linux, Unix on the individual physical machine are called virtu cluster nodes.

(ii) Virtual machine on physical machine but the operating syste of both are different. Virtual machine uses guest OS while physical machi uses host OS which manages resources of physical machine.

(iii) Use of virtual machines is to consolidate various functionaliti on the same server these features increase server utilization and flexibility application.

(iv) The number of nodes within a virutal cluster can increase decrease dynamically.

(v) When any physical node has failed. All virtual machines install on that mode will be disables but when vitaul machine has failed, host syste does not affect.

Q.17. What are the objectives of virtualization ?

Ans. The objectives of virtualization are –

(i) Improvement of scalability

(ii) Improvement of availability

(iii) Improvement of maintainability

(iv) Improvement of performance.

Q.18. What is management virtualization ?

[R.G.P.V., Dec. 2015 (MC

Ans. Virtualization management refers to coordinated provisioning orchestration of virtualized resources, as well as the runtime coordinati of resource pools and virtual instances. This feature includes the sta and dynamic mapping of virtual resources to physical resources, and a overall management capabilities such as capacity, analytics, billings, a SLAs.

Q.19. Write down the comparison between virtualization and cloud computing.

Ans. The comparison between virtualization and cloud computing are as follows –

S.No.	Virtualization	Cloud Computing
(i)	Part of the ordered substructure.	Brings resources of computing as a utility to client across the network.
(ii)	A self-service layer itself is not provided to the client and without that layer user cannot handover compute as utility.	Cloud deals computing as a service instead of a particular technology.
(iii)	One probable utility that can be delivered.	An access for the bringing of utilities to an clients.
(iv)	Can exist without the cloud.	Can exist only with virtualization.
(v)	Virtualization allows itself an arrangement to serve and efficiently use its IT resources	Using cloud computing it is possible to use those resources on other level by giving access to elements when required.

Q.20. Briefly explain the different types of virtualization.

[R.G.P.V., Nov. 2018 (BE)]

Ans. Following are the various types of virtualization –

(i) Execution Virtualization – Execution virtualization incorporates all those techniques whose objective is to emulate an execution environment that is separate from the one hosting the virtualization layer. All these techniques focus their interest on giving support for the programs execution. The programs may be an application, the operating system, or a binary specification of a program compiled against an abstract machine model. Thus, the implementation of execution virtualization can be done directly on top of the hardware, by an application, the operating system, or libraries statically or dynamically linked against an application image. Execution virtualization techniques are divided into two major categories depending on the type of host they need. The implementation of process level techniques are done on top of an existing operating system, which has full control of the hardware. The implementation of system level techniques are done directly on hardware and do not need an existing operating system.

(ii) Storage Virtualization – A system administration practice that enables decoupling the physical organization of the hardware from its logical representation is called storage virtualization. Users do not have to be thought

about the particular location of their data by employing this technique. The particular location of data is recognized through a logical path. Storage virtualization enables harnessing a variety of storage facilities and representing them under a single logical file system. We can divide the storage virtualization into different techniques. Among them, the most popular is network-based virtualization by means of Storage Area Networks (SANs). A network accessible device is used by storage area networks using a large bandwidth connection to offer storage facilities.

(iii) Network Virtualization – For the creation and management of a virtual network, network virtualization combines hardware appliances and specific software. Different physical networks can be aggregated into a single logical network (external network virtualization) by network virtualization. To an operating system partition (internal network virtualization), network virtualization offer network like functionality. Generally, the outcome of external network virtualization is Virtual LAN (VLAN). An aggregation of hosts that communicate with each other if they were located under the same broadcasting domain is a VLAN. Internal network virtualization is applied together with the hardware and operating system level virtualization where the guests achieve a virtual network interface to communicate with. The implementation of internal network virtualization are done in several ways – the guest can contain a private network only with the guest; the guest can share the same network interface of the host and use NAT to access the network; or the virtual machine manager can emulate, and install on the host, an additional network device together with the driver.

(iv) Desktop Virtualization – The desktop environment available on a personal computer is abstracted by desktop virtualization in order to give access to it by employing a client-server approach. Desktop virtualization makes accessible a different system, but this system is remotely stored on a different host and accessed using a network connection. Apart from this, desktop virtualization deals with the situation of making the same desktop environment accessible from everywhere. However, the term desktop virtualization means the ability to remotely access a desktop environment. In general, the desktop environment is stored in a data center or a remote server that offers a high availability infrastructure, and guarantees the accessibility and the persistence of the data. There are several benefits of desktop virtualization – accessibility, persistence, high availability and ease of management.

(v) Application-server Virtualization – A collection of application servers is abstracted by application-server virtualization. These application servers offer the similar services like a single virtual application server by employing load balancing strategies and offering a high availability infrastructure

for the services hosted in the application server. This is a specific type of virtualization and serves the same purpose of storage virtualization.

Q.21. What is the need of storage virtualization ? Discuss the ways in which it is structured. ***[R.G.P.V., Dec. 2017 (BE)]***

Ans. Storage virtualization needed for –

(i) Efficient storage utilization

(ii) Fast and reliable storage

(iii) Provides flexibility.

There are three ways in which storage virtualization structured –

(i) Host-based Virtualization – It is very simple way to provide storage virtualization. The responsibility of controlling physical storage space is depended on existing device drivers. A virtualization program on top of these drivers intercepts the input output request and gives the meta-data lookup and input output mapping.

(ii) Storage Device-based Virtualization – It is the other way of virtualization. In this way virtualization is performed on hardware level. It leverages the capability of RAID controllers. It manages meta-data by creating a logical storage space through the use of large number of physical device drivers resources. Advanced RAID controllers permit further storage devices to be attached as-well-as features like cloning and remote replication.

(iii) Network-based Virtualization – It is performed on network device like server. Network device resides between the host and storage gives the features of input output redirection, vitalizing input output requests and mapping between physical and logical space. Many storage device look like physically connected to the operating system. Storage and virtualization device and this network of hosts are known as storage area network.

Q.22. Write in brief about block and file level storage virtualization. ***[R.G.P.V., Dec. 2015 (BE)]***

Or

Write the difference between block and file level storage virtualization. ***[R.G.P.V., Dec. 2014 (BE), June 2015 (BE)]***

Ans. **Block Level Storage Virtualization** – Block level storage virtualization is illustrated in fig. 2.6. It offers a translation layer between the hosts and the storage arrays in the storage area network (SAN). Here, the hosts are directed to the virtualized logical unit numbers (LUNs) on the virtualization device. The translation between the virtual LUNs and the physical LUNs on the individual arrays is carried out by the virtualization device. This makes easy the use of arrays from several vendors at a time, without any interoperability issues. All the arrays seem as a single target device

and LUNs can be partitioned or distributed across multiple arrays for a host. Block level storage virtualization combines heterogeneous storage arrays, increases storage volume online, allows transparent volume access, and resolves application growth requirements.

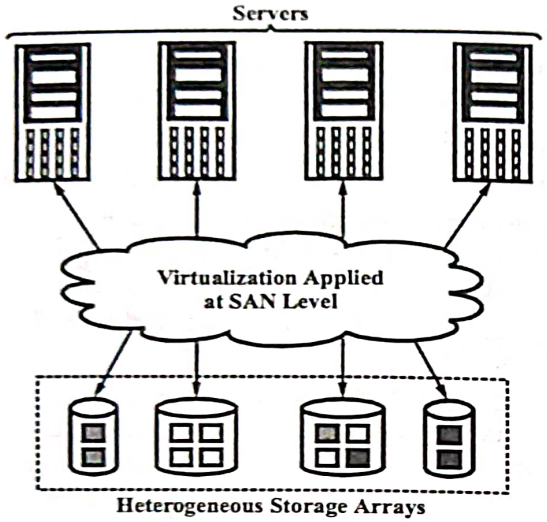
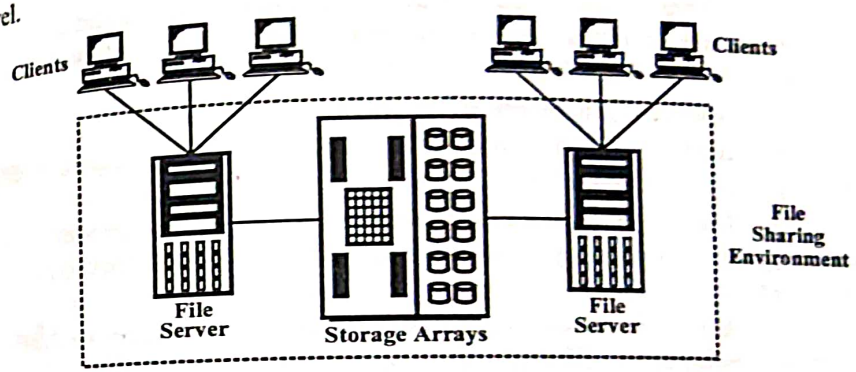


Fig. 2.6 Block Level Storage Virtualization

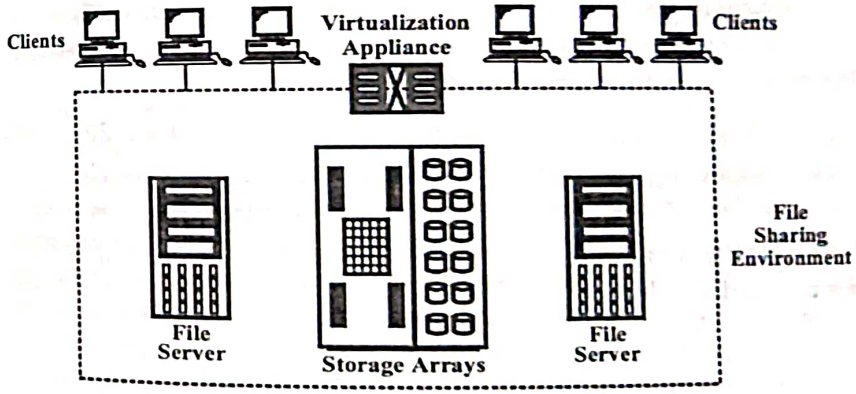
LUN migration from one array to some other array was an offline event in traditional SAN environments since the hosts required to be updated to reflect the new array configuration. In other cases, host CPU cycles were needed to migrate data from one array to the other, particularly in a multi-vendor environment. In case of block level virtualization solution, the virtualization engine manages the back-end migration of data that allows LUNs to remain online and accessible during data migration. Since the host still points to the same virtual targets on the virtualization device, no physical changes are needed. Although, there should be a change in the mappings on the virtualization device. These changes are transparent to the end user and can be executed dynamically. In a virtualized environment, deploying heterogeneous arrays makes easy an information lifecycle management (ILM) strategy, enabling considerable cost and resource optimization.

File Level Storage Virtualization – A network attached storage (NAS) environment before and after the implementation of file level virtualization is shown in fig. 2.7. File level virtualization offers opportunities to perform nondistruptive file migrations, and server consolidation and to optimize utilization. It meets the NAS challenges by alleviating the dependencies between the location where the files are physically stored and the data accessed at the file level. Each NAS device or file server is logically and physically independent

before virtualization. Each host knows exactly the location of its file level resources. Underutilized storage resources and capacity problems result due to boundation of files to a certain file server. Due to performance reasons or when the file server fills up, it is necessary to migrate the files from one server to another. It is not easy to move file across the environment and this needs downtime for the file servers. In addition, hosts and applications have to be reconfigured with the new path, making it complex for storage administrators to enhance storage efficiency at the time of maintaining the required service level.



(a) Before File Level Virtualization



(b) After File Level Virtualization

Fig. 2.7 NAS Device

The mobility of file is facilitated by file level virtualization. It offers users or application independence from the stored file location. File level virtualization generates a logical pool of storage, which allows users to use a logical path to access files. The movement of file systems across the online file servers is simplified by file level virtualization. It refers that when files are being moved, clients can access their files without causing problems. The files of clients

can be read from the old location and written them back to the differe location without realizing the change of physical location. Online moveme of files can be performed by multiple clients attached to multiple servers optimize utilization of their resources. The mapping of logical path of a file the physical path names can be done using a global namespace.

Q.23. Explain the importance of memory virtualization.

[R.G.P.V., Dec. 2017 (MCA)]

Ans. There are several importance of memory virtualization –

(i) Memory virtualization helps in memory utilization through sharing of resources which are scarce and consolidating more virtual machine on a physical host and sharing of contents.

(ii) Memory virtualization lowers latency.

(iii) Memory virtualization gives faster access as compare to SSD, SAN.

(iv) Memory virtualization helps in sharing large amount of data between applications of many server without redundancy and reduce memory requirement.

(v) Memory virtualization helps to access more memory as compared to capacity of physical memory.

(vi) Memory virtualization helps in reducing run time of input output bound applications which consume more data, thus increases efficiency.

(vii) Memory virtualization helps in keeping some memory space free before halting services as long as memory frees up.

Q.24. Describe application virtualization.[R.G.P.V., Dec. 2013 (MCA)]

Ans. Application virtualization is a method that describes software technologies that separate them from the underlying operating system on which they are executed. This applies to applications at all tiers, from servers to desktops. Generally, this is achieved using encapsulation. Encapsulating and isolating the application from the OS changes the way applications can install and interact with the OS. This makes it possible for the application to be moved from one system to another, patched, and updated without interactions with other applications. Application virtualization also allows applications to be copied or backed up as a single file. A fully virtualized application is not installed in the traditional sense, although it still executes as though it were. The application is tricked at run time to believe that it is directly interfacing with the original OS and the resources it manages.

Q.25. Discuss about virtualization applications in enterprises.

Ans. Some of the more compelling cases for using virtualization in large enterprises are as follows –

(i) Desktop Virtualization – In this approach, large enterprises have tens if not hundreds of thousands of users, each having a desktop and/or one or more laptops and mobile phones that are used to connect to applications running in the enterprise's data center. Managing regular system updates, such as for security patches or virus definitions is a major system management task. Sophisticated tools, such as IBM's Tivoli are used to automate this process across a globally distributed network of users. Managing application roll-outs across such an environment is a similarly complex task, especially in the case of 'fat-client' applications such as most popular email clients and office productivity tools, as well some transaction processing or business intelligence applications.

(ii) Application Streaming – In this approach, instead of running applications on central virtual machines, application streaming envisages maintaining only virtual machine images centrally. An end point client, such as a desktop, runs a hypervisor that also downloads the virtual machine image from the server and launches it on the end point client. In this manner the processing power of the end point is fully exploited, a VM image can be cached for efficiency and only incrementally updated when needed, and finally user data, which can be large, need not be centrally maintained but mounted from the local disk as soon as the virtual machine boots.

(iii) Security through Virtualization – Modern data centers are all necessarily connected to the world outside via the internet and are thereby open to malicious attacks and intrusion. A number of techniques have been developed to secure these systems, such as firewalls, proxy filters, tools for logging and monitoring system activity and intrusion detection systems. Each of these security solutions can be significantly enhanced using virtualization.

(iv) Server Consolidation – The most common driver for virtulization in enterprise data centers has been to consolidate applications running on possibly tens of thousands of servers, each significantly underutilized on the average, onto a smaller number of more efficiently used resources. The motivation is both efficiency as well as reducing the complexity of managing the so-called 'server sprawl.' The ability to run multiple virtual machines on the same physical resources is also key to achieving the high utilizations in cloud data centers.

(v) Automating Infrastructure Management – An important goal of enterprise virtualization projects is to reduce data center management costs, especially people costs through greater automation. It is important to recognize that while virtualization technology provides the ability to automate many activities, actually designing and putting into place an automation strategy is a complex exercise that needs to be planned.

Q.26. Enlist and explain some of the common pitfalls that come with virtualization. ***[R.G.P.V., Dec. 2013 (BE), June 2015 (BE), June 2017 (MCA)]***

Ans. Some of the pitfalls of virtualization are as follows –

(i) Inefficiency and Degraded User Experience – Sometimes, there is an inefficient use of host in virtualization. Some of the features of the host cannot be exposed by the abstraction layer and they become inaccessible. In case of hardware virtualization, this could happen when for device drivers the virtual machine can sometimes just provide a default graphic card which maps only a subset of the features available in the host. Some of the features of the underlying operating systems may become inaccessible unless specific libraries are used in case of programming level virtual machines.

(ii) Security Holes and New Threats – New and unexpected forms of phishing has introduced due to virtualization. The capability of emulating a host in a complete transparent manner, has led the way to malicious programs which are designed to extract sensitive information from the guest. Malicious programs can preload themselves before the operating system, and act as a thin virtual machine manager towards it in case of hardware virtualization. Then, the operating system is controlled, and can be manipulated in order to extract sensitive information of interest for third parties. BluePill and SubVir are the examples of these kind of malware. The diffusion of such kind of malware is facilitated by the fact that originally, hardware and CPU were not manufactured by keeping the virtualization in mind. The same considerations can be made for programming level virtual machines modified versions of the runtime environment can access sensitive information, or monitor the memory locations utilized by guest applications while these are executed. In order to make this possible, the original version of the runtime environment needs to be replaced by the modified one, and this can generally happen if the malware is run within an administrative context, or a security hole of the host operating system is exploited.

(iii) Performance Degradation – Increased latencies and delays can be experienced by the guest because virtualization interposes an abstraction layer between the guest and the host. For instance, in case of hardware virtualization, where the intermediate emulates a bare machine on top of which an entire system can be installed, the causes of performance degradation can be traced back by the overhead. Also, when hardware virtualization is realized through a program that is installed or executed on top of the host operating systems, a major source of performance degradation is represented by the fact that the virtual machine manager is executed and scheduled together with other applications, thus sharing with them the resources of the host.

Similar consideration can be made in case of virtualization technologies at higher levels, such as in the case of programming language virtual machines. The execution of managed applications is slow down due to binary translation and interpretation. Moreover, being their execution filtered by the runtime environment, access to memory and other physical resources can represent sources of performance degradation.

MULTITENANT SOFTWARE – MULTI-ENTITY SUPPORT, MULTI-SCHEMA APPROACH, MULTI-TENANCY USING CLOUD DATA STORES

Q.27. What is multi-tenancy ? Explain.

Ans. Multi-tenancy means sharing the application software between multiple users who have different needs. Allocating single instance of an application software i.e., cloud to multiple users is called as multi-tenancy. Each user is called as tenant. The users who need similar type of resources are allocated a single instance of cloud, so that the cost is shared between the users to make the access of instance of cloud computing cost effective. Multi-tenancy allows users to easily access, maintain, configure and manipulate the data stored in single database running on the same operating system. The data storage mechanism remains same for all users who share the similar hardware and software resources. In multitenant architecture, user cannot share or see each other's data, here the security and privacy is provided.

The main requirement of multi-tenancy is that the software provider gets many requests from customers with the customized needs. If a software product is implemented according to each customer needs separately and delivered, then the implementation takes more time to complete. The software cannot be maintained easily if there are different implementations of the product. The provider needs to spend more money to satisfy different customers. Here multi-tenancy comes into existence to provide solution for all the problems faced by provider to satisfy different customer with different needs. Multi-

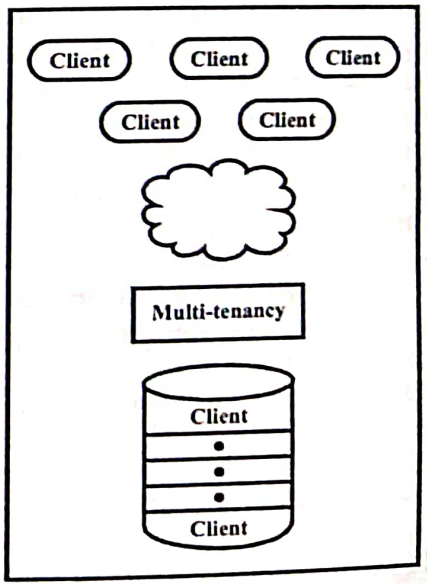


Fig. 2.8 Multi-tenancy

tenancy allows single software to be served between the multiple customers by using customized settings option. The needs of each customer are stored in custom settings. The software provider serves the same product by implementing it seeing the customized requirements of each customer and makes it available only to the specific customer respectively. The tenants who share the software product cannot see each other's implementation of product. There is no contact between each customer's sharing the same software. The software provider be in contact with multiple customers to satisfy them.

Q.28. Discuss about the multi-entity support.

Ans. Multi-tenancy is a way of trying to achieve an economic gain in cloud computing by utilizing virtualization and resource sharing. Multi-tenancy implies different meaning from different points of view and services. Means that,

To perform any type of services like IaaS, SaaS and PaaS in public cloud and private clouds the key technique is multi-tenancy. If the people discuss about the clouds they may speak about the IaaS services. Both cloud architectures like private and public clouds go beyond the special features like virtualization and the concept of IT-as-a-Service through payments or billing back in the event of private clouds based on metered usage. An IaaS service has advanced features such as service level agreements (SLAs), identity and access management for security access (IDAM), fault tolerance, disaster recovery, dynamic resource allocation and many other important properties. By injecting all these key services at the level of infrastructure, the clouds become multitenant to a degree. In the case of IaaS multi-tenancy go beyond the layer to merge the PaaS layer and at the end SaaS layer or application layer. IaaS layer contains servers, storages and networking components, PaaS layer consists of platform for applications like Java virtual machines like Java compilers, application servers and SaaS layer consists of applications like business logic, work flow, data bases and user interfaces.

The tenants can like the full stream of services that are commonly used from the cloud services, from the hardware infrastructure and going all the way up to the user interface based on the degree of multi-tenancy offered by the cloud. Cloud computing multi-tenancy is used for most if not all software as a service (SaaS) applications, because compute resources are scalable and allocation of these resources is defined by actual usage. There are different types of SaaS services that the clients can access by using internet, from low internet bases applications to a very big software applications that contains very high security requirements depends on the type of information stored on the software vendors infrastructure outside the corporate network.

There are basically two type of multi-tenancy techniques as follows –

(i) ***Virtual Multi-tenancy*** – In this computing and storage resources are shared among multiple users. Multiple tenants are served from virtual machines that execute concurrently on top of the same computing and storage resources.

(ii) ***Organic Multi-tenancy*** – In organic multi-tenancy every component i.e., hardware and software resources in the system architecture is shared among multiple tenants.

Software as a service provides a software model to deliver software based applications to provide remote access to the customers. In the cloud multi-tenancy is an important feature to provide SaaS services with different tenants simultaneously with a single application instance on the top of the shared infrastructure. Now-a-days SaaS applications are build with centralization through a single instance with multi-tenant architecture to provide an advance rich experience as compared to on-premise models. Advantage of multi-tenancy include operational costs are reduced by dividing hardware, software resources among the different tenants are shared, simplifying the maintenance and management effort. All of these advantages of multi-tenancy result in reducing the application costs to provide maximum benefits to small and medium organizations. Multi-tenancy service requirements for cloud services providers are tenant data isolation, tenant workspace isolation, isolation of tenant execution, tenant-aware security, monitoring, management, reporting and self-service administration, isolation of tenant customizations and extensions to business logic, tenant-aware version control, tenant-aware error tracking and recovery. The degree of multi-tenancy of an application is defined as the amount of base application or a SaaS layer is developed to be shared amount tenants. The highest degree of multi-tenancy allows the database schema to be shared and supports customization of the business logic, workflow and user-interface layers, private clouds are available at the lowest degree of multi-tenancy and are more suited for specific large enterprise customers.

Q.29. Discuss about the applications of multi-tenancy.

Ans. In multi-tenant applications, data and workload can be easily partitioned. We can partition data and workload along tenant boundaries because most requests occur within the confines of tenant. These SaaS applications deliver a specialized software application as a service to their tenants. Tenants can access the application service and have full ownership of associated data stored as part of the application. But to take advantage of the benefits of SaaS, tenants must surrender some control over their own data. They trust the SaaS service provider to keep their data safe and isolated

from other tenants' data. Examples of this kind of multi-tenant SaaS applicatio are MYOB, SnelStart and Salesforce.com. Each of these applications can b partitioned along tenant boundaries. Applications that provide a direct servic to customers or to employees within an organization (often referred to a users, rather than tenants) are another category on the multi-tenant applicatio spectrum. Customers subscribe to the service and do not own the data tha the service provider collects and stores. Service providers have less stringen requirements to keep their customers' data isolated from each other beyond government-mandated privacy regulations. Example of this kind of customer facing multi-tenant application are media content providers like Netflix, Spotify, and XboX LIVE. Other examples of easily partitionable applications are customer-facing, Internet-scale applications, or Internet of Things (IoT) applications in which each customer or device can serve as a partition. Partition boundaries can separate users and devices. All applications cannot be partitioned along a single property such as tenant, customer, user or device.

Q.30. What are the benefits of multi-tenancy ?

Ans. The benefits of multi-tenancy are as follows –

(i) Worry Free Capacity – Multi-tenancy provides companies of all sizes the ability to reside in the same infrastructure and data centre.

(ii) Lower Cost of Ownership – Because all users access their services from the same technology platform it is much easier to access automatic and frequent updates. No longer need to pay for report customizations or to add new functionalities.

(iii) API Integration Scalability – The integration of Web API is available in single-instances, but in the multi-tenancy environment, specific requests for integrations will now go into our product roadmap, and as they become available, they'll be rolled out to all customers.

(iv) Configurable to your Own Needs – This capability provides our customers with the ability to meet their requirements and communication styles to manage all IT and communication expenses.

(v) Access to the Latest Releases – Earlier rolling out a new update was a lengthy process because we had to code the change separately for each client instance to ensure that it was compatible with their customizations, perform QA, and then put the change into production. With more than 100 customers, it was a time-consuming task. But, now with multi-tenant environment, every customer's instance has the same base code, the roll-out of new releases is seamless and provide faster access to innovative features to manage IT and communication expenses.

Q.31. Discuss the various issues in multi-tenancy.

Ans. The various issues in multi-tenancy are as follows –

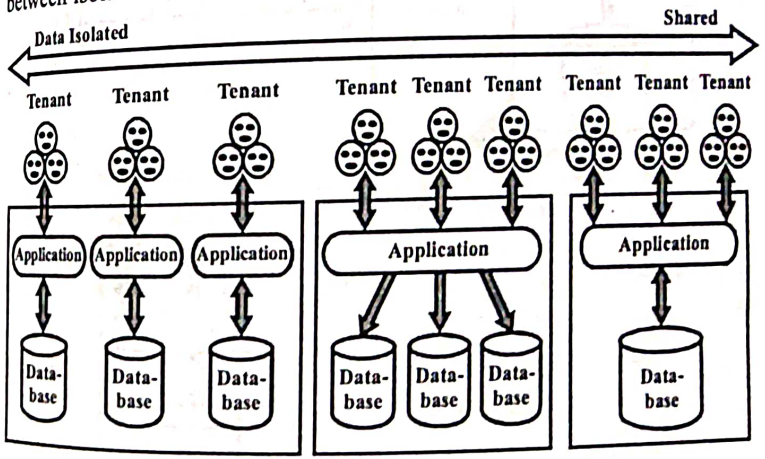
(i) Capacity Optimization – Database administrators need the tools and the knowledge to understand which tenant should be deployed on which network in order to maximize capacity and reduce costs.

(ii) Security – There is also the threat of hackers – no matter how secure an encryption is with the right knowledge. A hacker who breaks the encryption of multi-tenant database will be able to steal the data of hundreds of businesses who have data stored on it.

(iii) Service Delivery and High Availability – When failures occur or when certain services generate abnormal loads the service delivery can be interrupted – yet business client will often request high-availability. Therefore, monitoring the service delivery and its availability is critical to ensure that the service is properly delivered.

Q.32. Discuss about the multi schema approach.

Ans. The approaches towards building a multi-tenant database can be broadly categorized as fine-grained and coarse grained depending on the degree of multi-tenancy involved. There are three data storage strategies defined between isolated and shared features. These are shown in fig. 2.9.



(a) Separate Application, Separate Database *(b) Shared Application, Separate Database* *(c) Shared Application, Shared Database*

Fig. 2.9 Data Storage Strategies

(i) Separate Application, Separate Database [fig. 2.9 (a)] – In this model every user has its own software and database. All tenants are completely isolated from each other. Maintenance and update costs take too much time and system resources are not used efficiently in this model.

(ii) Shared Application, Separate Database [fig. 2.9 (b)] – In this model all tenants are using the same and only one software. Also, all users have their own physically separated databases. Special methods are used so that software can be individually customized for each tenant according to the tenant's wishes.

(iii) Shared Application, Shared Database [fig. 2.9 (c)] – In this model tenants use a common software. Also, this model is divided into two models. These are shared database-separate schema and shared database-shared schema. These are shown in fig. 2.10.

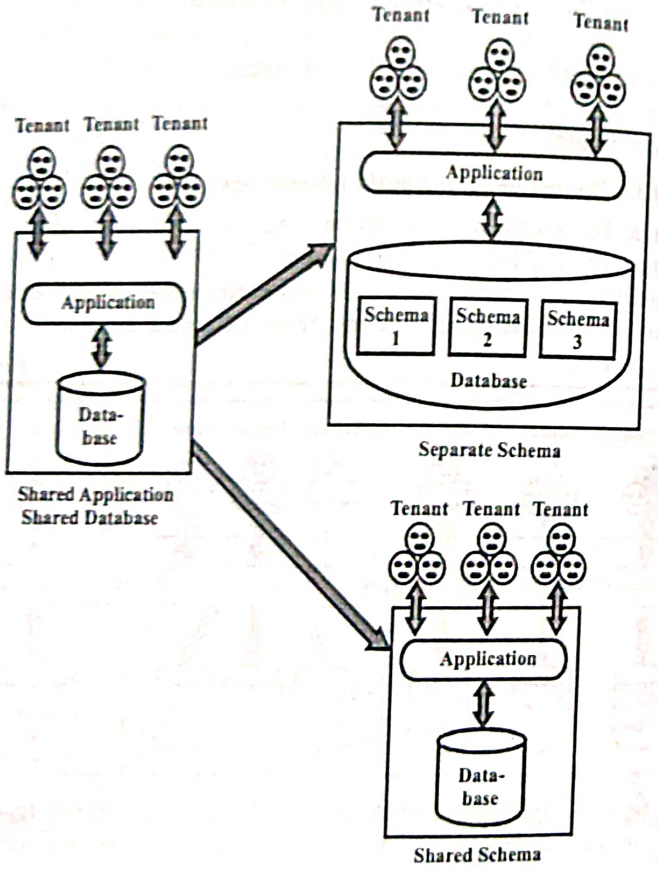


Fig. 2.10. Separate Schema, Shared Schema

(a) Shared Database, Separate Schema – In this approach, each tenant has its own separate database tables on the common database. Example is shown in fig. 2.11. As separate schemes are created for each tenant, the implementation is partly easier and can be easily shaped to suit the needs of a user. The disadvantage is that a large number of tables must be kept.

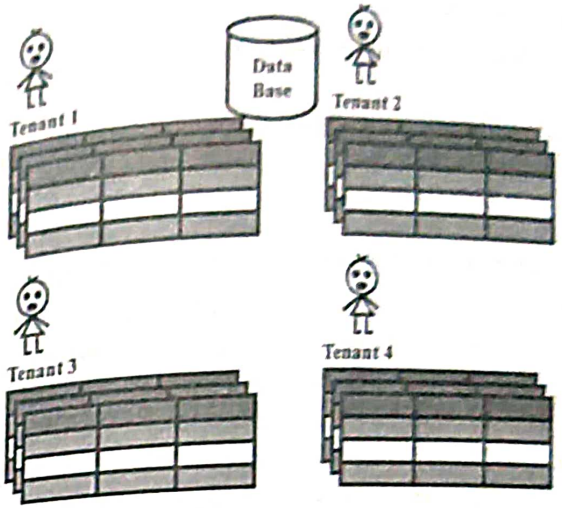


Fig. 2.11 Separate Schema

(b) Shared Database, Shared Schema – In this approach, there is a common database and common schema for all tenants. This model provides the least use and maintenance cost in terms of the most efficient use of hardware. The disadvantage is that all tenants are in the same schema.

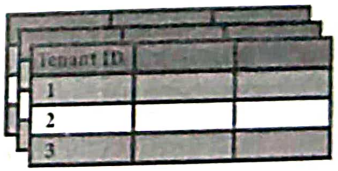


Fig. 2.12 Shared Schema

Q.33. Write short note on degree of multi-tenancy.

Ans. The degree of multi-tenancy is based on how much of a core application or SaaS, layer is designed to be shares across tenants. The highest degree of multi-tenancy allows the database schema to be shared and supports customization of the business logic, workflow and user-interface layers. In other words, all the sub-layers of SaaS offer multi-tenancy in this degree. The lowest degree corresponds to multi-tenancy only at infrastructure and platform level.

(i) Lowest Degree – IaaS and PaaS are multi-tenant, while SaaS is single tenant.

(ii) Middle Degree – IaaS and PaaS are multi-tenant. Small SaaS clusters are multi-tenant.

(iii) Highest Degree – IaaS and PaaS are multi-tenant. SaaS is fully multi-tenant also.

The higher the degree of multi-tenancy (meaning the more a cloud provider's infrastructure and resources are shared), the lower the costs for customers.

Q.34. Explain the multi-tenancy architecture in managing data.

Ans. Multi-tenancy is the defining characteristics of cloud computing. The shared infrastructure changes the underlying economies of enterprise applications, allowing vendors to maintain a single instance for thousands of customers. In multi-tenancy cloud environment, multiple users using the same infrastructure can access and use an application. The application design must therefore distinguish between users to ensure that they do not share each other's data. There are three different methods for achieving multi-tenancy which are – using a database, using virtualization and through physical separation. In the case of SaaS, multi-tenancy is achieved via database and configuration with isolation provided at the application layer. So at the application layer, service providers must design and implement a specific class and then create an object of the class in a manner that satisfies the need of multiple users in an effective way. Designing SaaS application in this way will solve many issues such as the need for data security, data separation and customized applications. Virtualization is another technology for achieving multi-tenancy especially for IaaS. Virtualization allows multiple copies of operations systems (VMs) run within a PM. These multiple VMs can then share the same physical hardware resources on the PM such as network card, disk storage. Though virtualization based multi-tenancy reduces costs and expenses, but compared to multi-tenancy using database technology, it is more costly.

Multi-tenancy can also be achieved through a dedicated technology that provides resources to tenants individually. This is known as multi-tenancy via physical separation. This option is by far the most expensive. Though CSPs offer this as an optional configuration to special customers, who want to use the cloud service but do not want to share hardware resources with other customers.

The three approaches to managing multi-tenant data in the cloud are – (i) storing tenant data on separate databases, which is the simplest approach to data isolation, (ii) housing multiple tenant on the same database, with each tenant having his own set of tables grouped into a schema created specifically for the tenant; and (iii) by using the same database, and same set of tables to host multiple tenants' data.

The general architecture for representing multi-tenancy for effective cloud environment is shown in fig. 2.13.

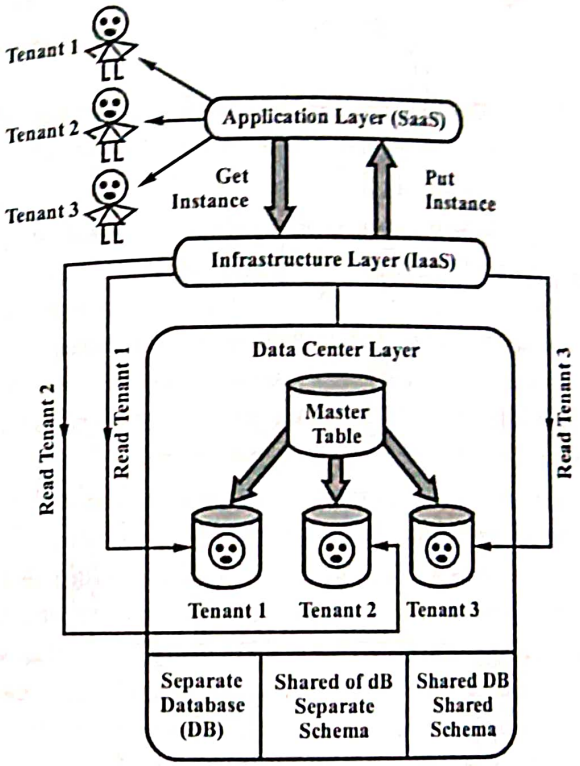


Fig. 2.13 Architecture of Multi-tenancy

The architecture employs customer integration in three layers, which are the application, the infrastructure and the data-centre layer.

The infrastructure layer and application layer consumer integration levels are latest additions to the cloud computing model. This integration is used to reduce the cost and need of developing highly scalable SaaS applications, which they do by compromising on security and customer segregation requirements.

(i) Data Centre Layer – This configuration provides the highest level of security requirements if implemented correctly, with firewalls and

access controls to meet business requirements as well as defined security access to the physical location of the infrastructure providing the SaaS. Mostly data centre layer multi-tenancy acts as a service provider that rents cages to companies that host their hardware, network, and software in the same building.

(ii) Infrastructure Layer – In infrastructure layer multi-tenancy the software stacks are provided. Each customer or tenant is provided with a dedicated software stack. This configuration saves costs compared to data centre-layer multi-tenancy, because stacks are deployed based on actual customer accounts.

The high availability of hardware and software resources can be seen in this layer. In this case, you can grow hardware requirements based on actual service use.

(iii) Application Layer – Application-layer multi-tenancy requires architectural implementations at both the software layer and the infrastructure layer. Modifications are required for the existing software architecture, including multi-tenant patterns in the application layer. For example, multi-tenant applications require application methods and database tables to access and store data from different user accounts, which compromises on security. If done accurately, however, the benefit is cost savings.

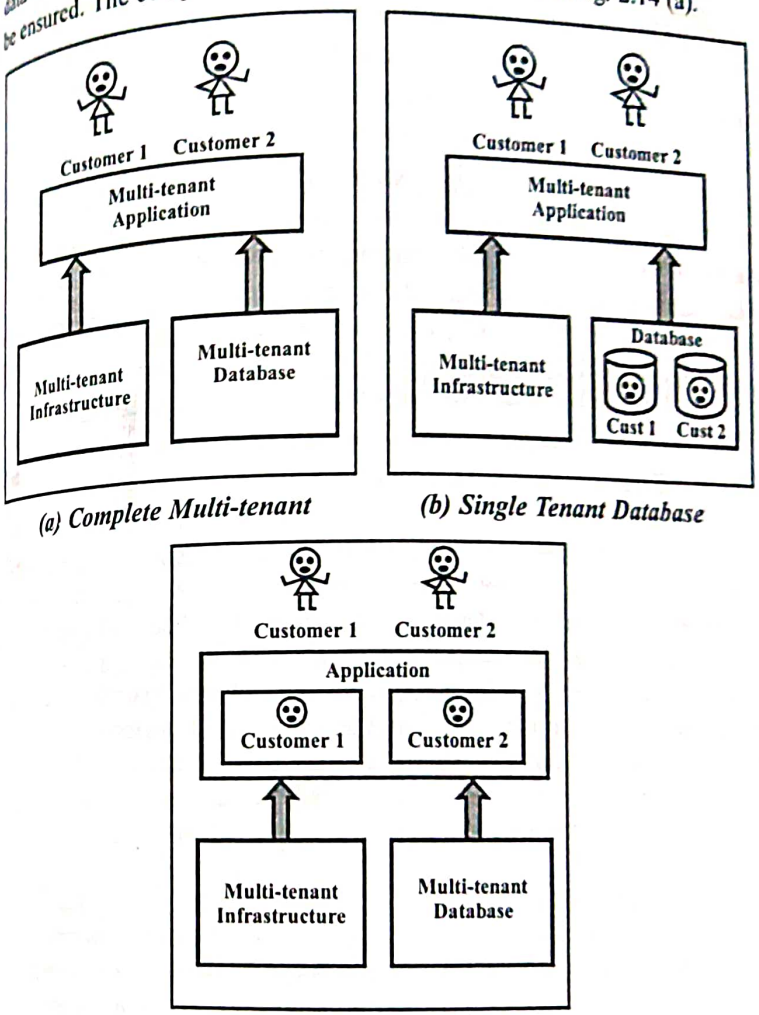
While multi-tenancy, on cloud environments provides seemingly limitless scalability and an alternative to the expensive data centre infrastructure, it raises security and privacy issues because it hands the processing and storage task over to third parties. This requires building adequate security into every aspect of the SaaS application, as well as for every IaaS virtual service. This involves using filtering which provides an intermediary layer between a tenant and data source. Another is permission which uses access control lists. Finally is encryption which obscures every tenant's critical data.

Q.35. Explain models for multi-tenant database.

Ans. Based on approach used for data isolation and other parameters related to tenants, we can have one of the following three basic models as shown in fig. 2.14.

(i) Complete Multi-tenant – This has highest degree of multi-tenancy by employing shared-everything approach. The inherent risks/complexities of this model include requirement of a very complex architecture, high business risk as the data is shared between all tenants, providing customized

data backup/restore services is not easy and equal distribution of load cannot be ensured. The complete multi-tenant model is shown in fig. 2.14 (a).



(a) Complete Multi-tenant *(b) Single Tenant Database*

(c) Single Tenant Application

Fig. 2.14 Models for Multi-tenancy at Database

(ii) Single Tenant Database – This has a moderate degree of multi-tenancy where application layer is commonly shared among all the tenants but not the databases. The level of separation is determined by the tenant architecture.

(a) Different Schemas within the Same Database – Where each schema can contain the tables for the respective tenant.

(b) Different Databases within the Same Machine – Each tenant gets a different database. This gives the flexibility to provide specific services like customized back-up and restore services.

(c) Different Databases Residing on Different Machines – Independent machines are allocated for each tenant. This provides the highest level of security from a data isolation/storage perspective.

(iii) Single Tenant Application – This is the inverse of the previous model. Here the database layer is kept common across all the tenants but the application layer is isolated. This model is employed where certain operations like customization of interface (in the form of change in business logic, rules, display, etc.) is to be performed over the application layer. The single tenant application model is shown in fig. 2.14 (c).

UNIT 3

DATA IN THE CLOUD – RELATIONAL DATABASES, CLOUD FILE SYSTEMS – GFS AND HDFS, FEATURES AND COMPARISONS AMONG GFS, HDFS, etc., BIGTABLE, HBASE AND DYNAMO

Q.1. How data is managed in the cloud ? Explain.

Ans. Data is collected in the form of an organized structure called the database which is the food of any information system. Data in huge amount is the major component of the cloud infrastructure. Data can be shared among many tenants. As a result, data management in particular is a key aspect of storage in the cloud. Data in the cloud is distributed across multiple sites and may contain certain privileges and authentic information. It is therefore very important to ensure that data consistency, scalability and security are maintained. In order to address these issues and many other important data issues, there is a need for a database management system for cloud data. The database management system shows the mechanism of storage and retrieval of user data with maximum efficiency, taking into consideration the appropriate security policies. The database management system always provides data independence. No change is made to the storage mechanism and shapes without modifying the entire application. There are several types of database organization, relational database, flat database, object oriented database, hierarchical database.

Structured data work with relational databases while non-relational databases work with semi-structured data. The non-relational database is known as (No-SQL). This category of database has been steadily adopted in recent years with the emergence of big data applications, since the purpose of designing non-relational databases is to overcome the limitations of relational databases in dealing with big data demands. Big data refers to data that is growing and moving very rapidly and is very diverse in the structure of traditional technologies to deal with. The difference between relational data and (No-SQL) is that the relational data model consists of a set of interconnected tables through keys, while (No-SQL) is increasingly considered a viable alternative

to relational databases, especially for big data applications. There are several database management systems in the computed cloud that provide storage and analysis for both relational (SQL) and non-relational (No-SQL). But No-SQL big data systems are designed to take advantage of new cloud computing structures, which makes big operational data much easier to manage, cheaper and faster to implement.

Q.2. Explain the relational cloud architecture.

Ans. The cloud database is constructed by collecting a number of sites. The sites are also called as nodes which are interlinked by a communication network. Every single node is a database class. Each database class has its own database, terminals, the central processor and their individual local database management system.

A relational database management system (RDBMS) is a software package with computer programs that controls the creation, maintenance, and use of a database. It allows the organizations to conveniently develop databases for various applications. A database is an integrated collection of data records, files and other objects. A DBMS allows different user application programs to concurrently access the same database. DBMSs may use a variety of database models, such as the relational model or object model to conveniently describe and support applications.

Relational cloud uses existing unmodified DBMS engines as the back-end query processing and storage nodes. Each back-end node runs a single database server. The set of back-end machines can change dynamically in response to load. Each tenant of the system which we define as a billable entity (a distinct user with a set of applications, a business unit, or a company) can load one or more databases. A database has one or more tables, and an associated workload, defined as the set of queries and transactions issued to it (the set may not be known until run-time). Relational cloud does not mix the data of two different tenants into a common database or table, but databases belonging to different tenant's will usually run within the same database server.

Applications communicate with relational cloud using a standard connectivity layer such as JDBC. They communicate with the relational cloud front-end using a special driver that ensures their data is kept private (e.g., cannot be read by the database administrator). When the front-end receives SQL statements from clients, it consults the router, which analyzes each SQL statement and uses its metadata to determine the execution nodes and plan. The front-end coordinates multi-node transactions, produces a distributed execution plan, and handles fail-over. It also provides a degree of performance isolation by controlling the rate at which queries from different tenants are dispatched.

The front-end monitors the access patterns induced by the workloads and the load on the database servers. Relational cloud uses this information to periodically determine the best way to – (i) partition each database into one or more pieces, producing multiple partitions when the load on a database exceeds the capacity of a single machine (ii) place the database partitions on the back-end machines to both minimize the number of machines and balance load, migrate the partitions as needed without causing downtime, and replicate the data for availability (iii) secure the data and process the queries so that they can run on untrusted back-ends over encrypted data. The relational cloud system architecture is shown in fig. 3.1 which depicts these functions and demarcates the trusted and untrusted regions.

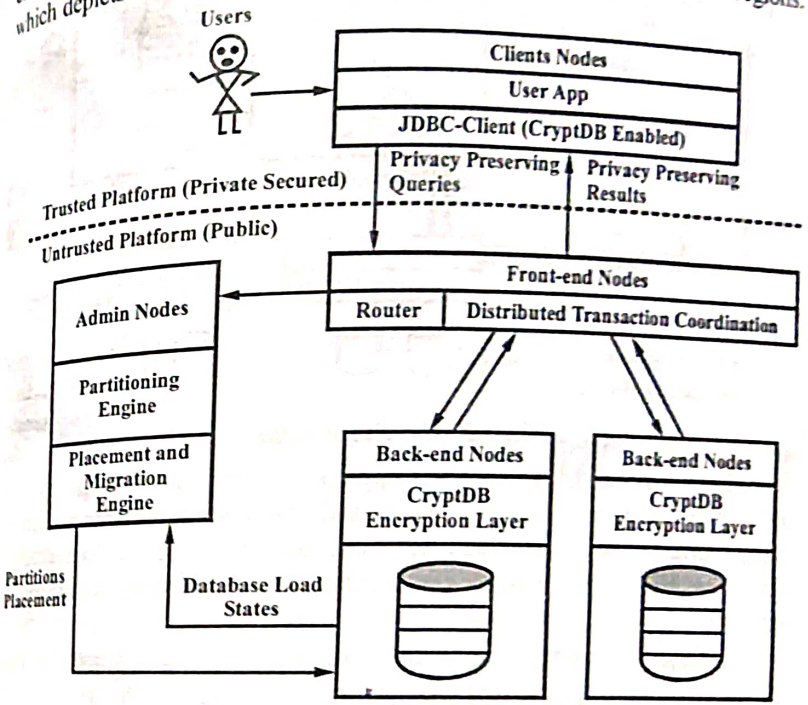


Fig. 3.1 Relational Cloud Architecture

Q.3. Explain the architecture of cloud file systems (GFS, HDFS).
[R.G.P.V., June 2017 (MCA)]

Ans. **Architecture of GFS** – A GFS cluster consists of a single master and multiple chunk servers and is accessed by multiple clients, as shown in the fig. 3.2.

Each of these is typically a commodity Linux machine running a user-level server process. Files are divided into fixed-size chunks. Each chunk is identified by a fixed and globally unique 64-bit chunk handle assigned by the master at the time of chunk creation. Chunk servers store chunks on local disks as Linux files. For reliability, each chunk is replicated on multiple chunk

servers. By default, there will three replicas and this value can be changed by user. The master maintains all file system metadata. This includes the namespace, access control information, the mapping from files to chunks, and the current locations of chunks. It also controls system-wide activities such as chunk lease management, garbage collection of orphaned chunks, and chunk migration between chunk servers. The master periodically communicates with each chunk server in Heart Beat messages to give it instructions and collect its state.

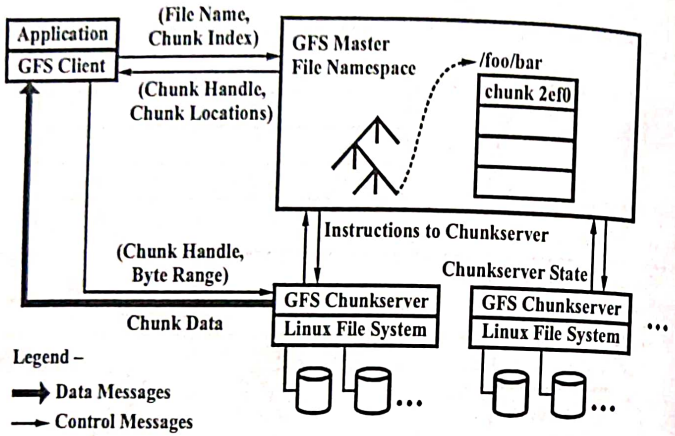


Fig. 3.2 GFS Architecture

GFS client code linked into each application implements the file system API and communicates with the master and chunk servers to read or write data on behalf of the application. Clients interact with the master for metadata operations, but all data-bearing communication goes directly to the chunk servers.

Architecture of HDFS – HDFS is the master/slave structure. The Namenode is the master node, while the Datanode is the slave node. Documents are stored as data blocks in the Datanode. The default size of a data block is 64M and it cannot be changed. If the files are less than a block data size, HDFS will not take up the whole block storage space. The Namenode and the Datanode normally run as Java programs in the Linux operating system.

The Namenode which is the manager of the HDFS is responsible for the management of the namespace in the file system. It will put all the folders and files metadata into a file system tree which maintains all the metadata of the files directories. At the same time, Namenode also saves the corresponding relations between each file and the location of the data block. Datanode is the place to store the real data in the system. However, all the data is not stored on

the hard drives but will be collected when the system starts to find the resource data server of the required documents.

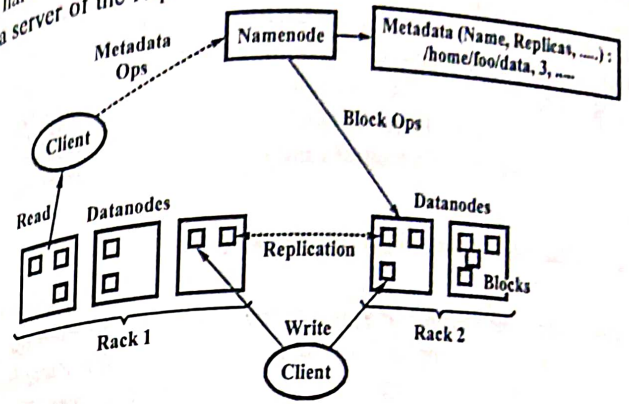


Fig. 3.3 HDFS Architecture

The Secondary Namenode is a backup node for the Namenode. If there is only one Namenode in the Hadoop cluster environment, the Namenode will obviously become the weakest point of the process in the HDFS. Once the failure of the Namenode occurs, it will affect the whole operation of the system. This is the reason why Hadoop designed the Secondary Namenode as the alternative backup. The Secondary Namenode usually runs on a separate physical computer and keeps communication at certain time interval to keep the snapshot of the file system metadata with the Namenode so that it can recovery the data immediately in case some error happens.

The Datanode is the place where the real data is saved and handles most of the fault-tolerant mechanism. The files in HDFS are usually divided into multiple data blocks stored in the form of redundancy backup in the Datanode. The Datanode reports the data storage lists to the Namenode regularly so that the user can obtain the data by directly access to the Datanode.

The client is the HDFS user. It can read and write the data though calling the API provided by HDFS. While in the read and write process, the client first needs to obtain the metadata information from the Namenode, and then the client can perform the corresponding read and write operations.

Q.4. Write about the Google file system ?

Ans. Google File System (GFS) was designed and implemented in early 2002 and is considered a vital part of the GAE. GFS is a file system designed to support searching and web crawling. GFS is a proprietary distributed file system developed by Google keeping in mind their own application workloads and technological environment designed to be fault tolerant on inexpensive commodity hardware serving a large number of

clients. Google uses this as the storage platform for the data generation and processing used by their services as well as research efforts that require large data sets. This set-up includes data of the order of hundreds of terabytes stored over thousands of disks covered by thousands of machines. GFS is probably the largest file system in the world that is in operation. It is highly efficient and scalable. Google has reported that the number of users using Google has grown significantly over time, and sometimes at a rapid rate.

Q.5. What are the features of Google file system ?
[R.G.P.V., May 2019 (MCA)]

Ans. Google file system (GFS) is a distributed file system built for large distributed data intensive applications like gmail etc. Originally, it was built to store data generated by its large crawling and indexing system. The files generated by this system were usually huge. Maintaining and managing such huge files and data processing demands was a challenge with the existing file systems. The main objective of the designers was building a highly fault tolerant system while running inexpensive hardware.

The features of Google file system are as follows –

(i) Fault tolerance.
(ii) Critical data replication.
(iii) Automatic and efficient data recovery.
(iv) High aggregate throughput.
(v) Reduced client and master interaction because of large chunk server size.
(vi) Name space management and locking.
(vii) High accessibility.

The largest GFS clusters have more than 1000 nodes with 300 TB disk storage capacity.

Q.6. Discuss in detail about Hadoop Distributed File System (HDFS).
(R.G.P.V., May 2019)

Ans. HDFS also known as Hadoop Distributed File System is one the Hadoop components which handles the storage of big data. When users need to add more storage in the system, then they can easily increase the storage capacity by adding servers. HDFS consist of number of clusters depending upon the user configurations. The cluster consists of Master and Slave nodes. The data in the Hadoop cluster are broken into many small blocks which are 128 MB sizes by default. These blocks are stored in the different slaves' nodes in the Hadoop clusters. These blocks are highly scalable and can be increased when needed.

In HDFS, users can create new file, append content to the end of file, delete or rename the file, and modify file attributes. In comparison to traditional method of handling data, Hadoop's storage can be scalable at a very low cost because Hadoop uses commodity hardware.

Hadoop is composed of clusters. And cluster have Master node and Slave node as shown in the fig. 3.4. Master node is also known as name node which assigns jobs to the slave nodes. Beside assigning jobs to the slave nodes, master node manages the file system namespace. All the details are store in the form of namespace image and edit log. Cluster have only one master node, where as it may have multiple slave nodes. The function of slave nodes is to store data in the form of blocks and performed a job assigned by the master node.

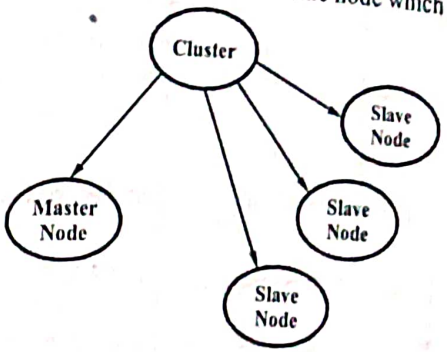


Fig. 3.4 Hadoop Cluster Node

HDFS is a distributed system which is suitable for running on the commodity hardware. There are many common characteristics in the existing distributed systems but the differences between them are also obvious. HDFS is a high fault-tolerant system and relaxed the parts of the POSIX constraints to provide high throughput access to the data so that it can be suitable to applying on the big data.

Q.7. What are the features of HDFS ? List out the characteristics of HDFS.

Ans. **The Features of HDFS** – HDFS is not a general-purpose file system, as it only executes specific types of applications, it does not need all the requirements of a general distributed file system. For example, security has never been supported for HDFS systems.

Characteristics – The characteristics of HDFS are as follows –

(i) HDFS fault tolerance
(ii) Block replication
(iii) Relica placement
(iv) Heartbeat and block report messages
(v) HDFS high throughput access to large dataset.

Q.8. Give a comparison between GFS and HDFS.

Ans. The comparison between GFS and HDFS is shown in table 3.1.

Table 3.1

S.No.	Properties	GFS	HDFS
(i)	Design goals	GFS stores a huge number of files, totaling many terabytes of data.	HDFS has to provide high aggregate data bandwidth and it has to scale to hundrads of notes in a single cluster. It is designed more for batch processing rather than interactive use by users.
(ii)	Processes	Master and chunk server.	Name node and data node.
(iii)	File management	In GFS, files are organized hierarchically in directories and identified by path names. It is support google file system only.	HDFS supports a traditional hierarchical file organization. It also supports third party file system such as cloud store and Amazon simple storage services.
(iv)	Database files	Bigtable is the database used by GFS.	Hbase provides bigtable like capabilities on top of Hadoop core.
(v)	Security	Google has dozens of datacenter for redundancy. These datacenters are in undisclosed locations and most are unmarked for protection.	HDFS security is based on the POSIX model of users and groups.
(vi)	Communication	TCP connections are used for communication. Pipelining is used for data transfer over TCP connections.	RPC based protocol on top of TCP/IP.

Q.9. What is bigtable ? Describe the main features of bigtable.

Ans. Bigtable is Google's proprietary distributed database management architecture designed for managing structured data while providing high scalability and performance. Many of Google's own services, such as Google Earth, Google Analytics, Orkut, and web indexing, utilize bigtable as the underlying data storage. Bigtable is also designed with flexibility in mind to cope with the very diverse demands placed on the system data size, ranging from URL (uniform resource locator) strings to satellite imagery.

Bigtable is a "sparse, distributed, persistent multi-dimensional sorted map" that is indexed by a row key, a column key, and a timestamp. The table is ordered by a row key and partitioned dynamically by a row range into tablets, which are the units of distribution in database load balancing. The row keys can be arbitrary strings and have a maximum size of 64 KB, and the operations on row keys are atomic. The row key range partitioning can be utilized further in the database and client design to limit the communication caused by shorter range queries to a small amount of machines in the cluster.

Bigtable uses Google file system (GFS) as a storage platform. GFS is a scalable and fault tolerant distributed file system, which works with huge files (by traditional standards). To achieve reliability and availability, files are divided into chunks, and each chunk is replicated across multiple machines.

The main features of Bigtable are as follows –

(i) Original Data Model – Bigtable is a sparse, distributed, persistent multi-dimensional sorted map. The map is indexed by a row key, column key, and a timestamp, and ordered by a row key. Client applications can access data either by row key or by range of row keys.

(ii) Strong Consistency – Bigtable stores data in GFS as immutable files (SSTables) and does not replicate data in bigtable level (as opposed to the GFS), therefore it can be considered as strongly consistent.

(iii) Single-master – Bigtable, as well as GFS, is designed with a centralized approach – a single master node maintains all system metadata. This architecture significantly simplifies system's design.

Q.10. Explain the architecture of bigtable.

Ans. Bigtable has three main components, one master server, several tablet servers and clients. Depending on the workload tablet servers are added and removed from a cluster dynamically. Master server is in charge of assigning tablets to tablet servers, balancing tablet server load, garbage collection and detection, addition and expiration of tablet servers. Besides that it manages changes in the schema (ex. table and column family creation). Every tablet server manages several tablets (between ten to thousand tablets per tablet server). Tablet server splits tablets that have grown too big and manages write and read requests for loaded tablets.

Bigtable is a single-master distributed storage system. In order for clients to perform reads and writes they have to communicate directly with tablet servers and most of the clients never communicate with the master because bigtable clients do not obtain tablet location information from the master. Therefore, the reality is that the master is not loaded most of the time.

Table 3.1

S.No.	Properties	GFS	HDFS
(i)	Design goals	GFS stores a huge number of files, totaling many terabytes of data.	HDFS has to provide high aggregate data bandwidth and it has to scale to hundrads of notes in a single cluster. It is designed more for batch processing rather than interactive use by users.
(ii)	Processes	Master and chunk server.	Name node and data node.
(iii)	File management	In GFS, files are organized hierarchically in directories and identified by path names. It is support google file system only.	HDFS supports a traditional hierarchical file organization. It also supports third party file system such as cloud store and Amazon simple storage services.
(iv)	Database files	Bigtable is the database used by GFS.	Hbase provides bigtable like capabilities on top of Hadoop core.
(v)	Security	Google has dozens of datacenter for redundancy. These datacenters are in undisclosed locations and most are unmarked for protection.	HDFS security is based on the POSIX model of users and groups.
(vi)	Communication	TCP connections are used for communication. Pipelining is used for data transfer over TCP connections.	RPC based protocol on top of TCP/IP.

Q.9. What is bigtable ? Describe the main features of bigtable.

Ans. Bigtable is Google's proprietary distributed database management architecture designed for managing structured data while providing high scalability and performance. Many of Google's own services, such as Google Earth, Google Analytics, Orkut, and web indexing, utilize bigtable as the underlying data storage. Bigtable is also designed with flexibility in mind to

cope with the very diverse demands placed on the system data size, ranging from URL (uniform resource locator) strings to satellite imagery.

Bigtable is a "sparse, distributed, persistent multi-dimensional sorted map" that is indexed by a row key, a column key, and a timestamp. The table is ordered by a row key and partitioned dynamically by a row range into tablets, which are the units of distribution in database load balancing. The row keys can be arbitrary strings and have a maximum size of 64 KB, and the operations to row keys are atomic. The row key range partitioning can be utilized further in the database and client design to limit the communication caused by shorter range queries to a small amount of machines in the cluster.

Bigtable uses Google file system (GFS) as a storage platform. GFS is a scalable and fault tolerant distributed file system, which works with huge files (by traditional standards). To achieve reliability and availability, files are divided into chunks, and each chunk is replicated across multiple machines.

The main features of Bigtable are as follows –

(i) Original Data Model – Bigtable is a sparse, distributed, persistent multi-dimensional sorted map. The map is indexed by a row key, column key, and a timestamp, and ordered by a row key. Client applications can access data either by row key or by range of row keys.

(ii) Strong Consistency – Bigtable stores data in GFS as immutable files (SSTables) and does not replicate data in bigtable level (as opposed to the GFS), therefore it can be considered as strongly consistent.

(iii) Single-master – Bigtable, as well as GFS, is designed with a centralized approach – a single master node maintains all system metadata. This architecture significantly simplifies system's design.

Q.10. Explain the architecture of bigtable.

Ans. Bigtable has three main components, one master server, several tablet servers and clients. Depending on the workload tablet servers are added and removed from a cluster dynamically. Master server is in charge of assigning tablets to tablet servers, balancing tablet server load, garbage collection and detection, addition and expiration of tablet servers. Besides that it manages changes in the schema (ex. table and column family creation). Every tablet server manages several tablets (between ten to thousand tablets per tablet server). Tablet server splits tablets that have grown too big and manages write and read requests for loaded tablets.

Bigtable is a single-master distributed storage system. In order for clients to perform reads and writes they have to communicate directly with tablet servers and most of the clients never communicate with the master because bigtable clients do not obtain tablet location information from the master. Therefore, the reality is that the master is not loaded most of the time.

Bigtable cluster stores several tables. Each table contains set of tablets and each tablet stores all the data associated with a row range. In the beginning, each table contains one tablet but with the growth of table it is split automatically into multiple tablets (default size is 100-200 MB for each). Other components of a bigtable system are GFS (persistence layer), Google MapReduce (for sorting tasks), Chubby (Meta data, configuration, distributed locking).

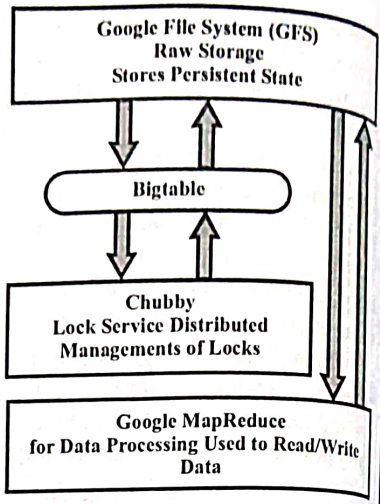


Fig. 3.5 Architecture of Bigtable

A bigtable cluster generally runs on a shared machine pool, often overlapping processes from other services, and is reliant of Google's other technologies. Bigtable operates on Google file system (GFS) and uses it to store log and data files. The data is stored in SSTable file format that provides a "persistent, ordered immutable map from keys to values", lookup operations for value retrieval by a key, and iterating over a range of key-value pairs.

Resource locking is performed by Chubby, a "highly-available and persistent lock service" utilizing the Paxos algorithm. Chubby is used by bigtable for following tasks –

(i) Master server management ensuring there is only one master at a time.

(ii) Discovering and finalizing the death of tablet servers.

(iii) Storing column family information (schema) for the bigtable instance.

(iv) Storing access control lists, and

(v) Storing bootstrap information location for bigtable data.

Q.11. Describe the main components of bigtable.

Ans. The three major components of bigtable are as follows –

(i) The Master Server – This is the first of the major components of the bigtable. The master server keeps track of the pool of available tablet servers and manages the server tablet machines by assigning tablets, handling column family schema alterations, detecting changes in the tablet server cluster, balancing load on the tablet server cluster, and handling garbage collection in the file system. The master server is not under a heavy load due to clients communicating directly with the tablet servers thus not requiring data to pass through the master server.

(ii) The Tablet Server – This, being the second of major components, handles a number of tablets and performs read and write operations and also divides tablets growing too large into smaller tablets.

(iii) The Client Library – The third of the major components, the client library, manages and directs requests to appropriate tablet servers and returns the responses back to the client application.

Q.12. What do you understand by Hbase ? Explain the architecture of Hbase.

Ans. Hbase is the Hadoop database which can provide real-time access to the data and powerful scalability. Hbase was designed based on the Bigtable, a database was launched by Google. Hbase aims at storing and processing big data easily. More specifically, it uses a general hardware configuration to process millions of data. Hbase is an open source, distributed, has multiple versions, and uses the NoSQL database model. It can be applied on the local file systems and on HDFS. In addition, Hbase can use the MapReduce computing model to parallel process big data in Hadoop. This is also the core feature of Hbase. It can combine data storage with parallel computing perfectly.

Architecture of Hbase – Hbase is the storage layer in the Hadoop. Its underlying storage support is HDFS, using the MapReduce framework to process the data, and cooperate with the ZooKeeper. The architecture of Hbase is shown in fig. 3.6.

The four key components are as follows –

(i) Hbase Client – The client is the user of the Hbase. It takes part in the manage operations with HMaster and read/write operations with HRegionServer.

(ii) ZooKeeper – ZooKeeper is the collaborative management node of Hbase. It can provide distributed collaboration, distributed synchronization, and configuration functions. The ZooKeeper coordinates all the clusters of Hbase by using data which contains the HMaster address and HRegionServer status information.

(iii) HMaster – HMaster is the controller of the Hbase. It is responsible for adding, deleting, and quering the data. It adjusts the HRegionServer load balance and the Region distribution to ensure that the region will move to the next Region when the HRegionServer suffers failure. An Hbase environment can launch multiple HMaster to avoid failure. At the same time, there is always a Master Election mechanism working in case of the node failure.

(iv) HRegionServer – HRegionServer is the core component of Hbase. It is responsible for handling the reading and writing requests for the users and performing the corresponding operations on HDFS.

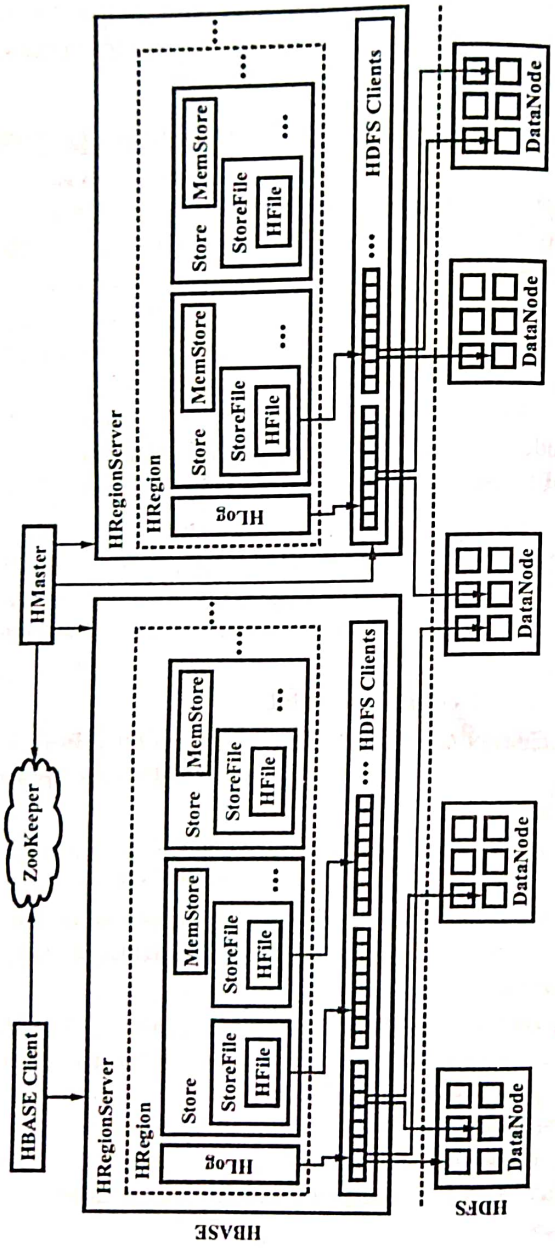


Fig. 3.6 Hbase Architecture

Q.13. Give the comparison of RDBMS and Hbase.

Ans. The comparison of RDBMS and Hbase is as follows –

Table 3.2

S.No.	Criteria	Hbase	RDBMS
(i)	Changeable data	Yes	Yes
(ii)	Data layout	A sparse, distributed, persistent multidimensional sorted map.	Row-oriented or column-oriented.
(iii)	Data types	Bytes; data types are interpreted on query.	Rich data type support.
(iv)	Hardware	Hadoop-clustered commodity x86 servers; five or more is typical because the underlying storage technology is HDFS, which by default requires three replicas.	Typically large, scalable multi-processor systems.
(v)	High availability	Yes; built into the Hadoop architecture.	Yes, if the hardware and RDBMS are configured correctly.
(vi)	Indexes	Row-key only or special table required.	Yes
(vii)	Query language	Hbase API commands (get, put, scan, delete, increment, check), HiveQL	SQL

Hbase, as the representative database, is often compared with the traditional RDBMS. The design target, implementation mechanism, and running performance are different. Due to the reason that the Hbase and RDBMS can replace each other in some special situations, it is inevitable to compare RDBMS with Hbase. As mentioned before, Hbase is a distributed database system and the underlying physical storage uses the Hadoop distributed file system. It does not have particularly strict requirements on the hardware platform. However, RDBMS is a fixed structure database system. The difference between their design goals makes them have the greatest difference in the implementation mechanism.

Q.14. What are the differences between bigtable and Hbase ?

Ans. The difference between bigtable and Hbase are given in table 3.3.

Table. 3.3

S.No.	Bigtable	HBase
(i)	At new master start-up tablet server list is read from Chubby and master asks for metadata to tablet servers.	Zookeeper does not have such function, region servers report metadata to masters themselves.
(ii)	Bigtable uses microseconds.	HBase stores timestamps in milliseconds.
(iii)	Bigtable relies on GFS (Google File System).	HBase works with Hadoop Distributed File System (HDFS). It can also run on other file systems.
(iv)	Bigtable can store files into memory.	HBase cannot store files into memory.
(v)	Bigtable implements a key/value cache, has two commit logs and is able to select which one to use.	HBase has an option to skip the commit log completely on writes for performance reasons.
(vi)	Bigtable can memory-map entire storage files and use them to perform lookups without a single disk seek.	HBase has an in-memory option per column family and uses its LRU cache to retain blocks for subsequent use.

Q.15. What is dynamo ? Give its key features.

Ans. Dynamo is a collection of key-value pairs that are indexed and retrieved by the primary key, providing only simple read and write operations to the data item uniquely identified by the key. The lightweight nature of the data structure and operations help key-value stores gaining performance, but it comes with the cost of the more advanced features such as complex querying. Due to key-value stores tying all data manipulation strictly to the keys, the rest of the data is opaque to the query. In comparison to other types of stores, the simplicity of the data model grants higher performance, but at the same time prevents any operations to non-key fields. Similarly, due to the database operations' single-key restriction, operations spanning multiple items are not supported. The schemaless data store is designed for applications and services with relative small data item sizes, usually less than 1 Mi.

The key dynamo's features are as follows –

(i) Key-value Data Model – Data re stored as objects which are identified by unique keys. Only two operations on data are supported – get/put value with the specified key. This simple data model was chosen after the observation that a significant portion of Amazon's services does not need a more complicated schema.

(ii) Symmetry and Decentralization – Dynamo is a completely decentralized system where every node has the same set of responsibilities as its peers. Thus, there is no single point of failure and only minimal manual administration is needed.

(iii) Eventual Consistency – Dynamo's main goal is to be highly available, therefore, according to the CAP theorem, it cannot be fully consistent. Instead, it provides an eventual consistency, that is, all updates reach all replicas eventually.

Q.16. Write down the comparison between dynamo and bigtable.

Ans. The comparison between dynamo and bigtable is given in table 3.4.

Table 3.4

S.No.	Category	Dynamo	Bigtable
(i)	Replication	Across data centers	No replication in Bigtable. Replication in GFS within a single data center.
(ii)	Storage	Plug-in	SSTables in GFS
(iii)	Read/ Write	Quorum-like	Reads – merge of SSTables and memtable Writes – tablet log and memtable
(iv)	Concurrency Control	Vector clocks with reconciliation during reads	Copy-on-write
(v)	Membership and failure detection	Gossip-based protocol	Handshakes initiated by master
(vi)	CAP	AP	CP
(vii)	Architecture	Decentralized	Centralized
(viii)	Data model	Key-value	Column-oriented
(ix)	API	get, put	get, put, scan, delete
(x)	Security	No	Access control at column-family level
(xi)	Partitioning	Consistent hashing	Key range based

MAPREDUCE AND EXTENSIONS – PARALLEL COMPUTING, THE MAPREDUCE MODEL – PARALLEL EFFICIENCY OF MAPREDUCE, RELATIONAL OPERATIONS, ENTERPRISE BATCH PROCESSING, EXAMPLE/APPLICATION OF MAP-REDUCE

Q.17. Explain the concept of MapReduce.

[R.G.P.V., Dec. 2014 (MCA), 2015 (MCA)]

Or

Explain the concept of MapReduce in cloud management.

[R.G.P.V., Dec. 2016 (BE)]

Ans. The concept of MapReduce is introduced by Google. MapReduce is a programming platform to process huge amount of data. Here, map and reduce are the two simple functions used to represent the computation logic of an application. The distributed storage infrastructure completely handles the data transfer and management. Distributed storage infrastructure is in-charge of providing access to data, replicating files and finally moving them where required. Hence, these issues are not handled anymore by developers and are offered with an interface which offers data at a higher level like a collection of key-value pairs. Then, the MapReduce applications computation is arranged in a workflow of map and reduce operations. These operations are fully managed by the runtime system, and developers have only to mention how these operations work on the key-value pairs. The MapReduce programming model is represented in the form of the two functions – map and reduce. These functions are written as –

map (k1, v1) → list (k2, v2)

reduce (k2, list (v2)) → list (v2)

A list of different key-value pairs is generated by the map function after reading the given key-value pairs. Then, a pair composed by a key and a list of values is read by the reduce function that generates a list of values of the same type. Here, the output of map function is aggregated together by combining the values on the basis of their corresponding keys and forms the input for reduce function. In reduce function, the list of attached values is reduced to a single value for each of the keys found. Thus, the MapReduce computation input is represented as a collection of key-value pairs and the output is represented by a list values.

Q.18. Explain some of the variations and extensions of MapReduce.

Ans. A number of variations and extensions to the original MapReduce model have been proposed due to its limitations. Their objective is to extend

MapReduce application space and offers an easier interface to developers for designing distributed algorithms. Some of the MapReduce like framework are as follows –

(i) Hadoop – A group of software projects for reliable and scalable distributed computing is Apache Hadoop. The complete group is an open-source implementation of the MapReduce framework aided by a GFS-like distributed file system. The initiative comprises two projects, namely, Hadoop MapReduce and Hadoop Distributed File System (HDFS). HDFS is an implementation of the Google File System. Hadoop MapReduce offers the similar features and abstractions of Google MapReduce. Hadoop was initially developed and supported by Yahoo. Now, it constitutes the most mature and large data cloud application. Hadoop has a very robust community of developers and users that support it. The world's largest Hadoop cluster has been composed by 40000 machines and more than 300000 cores. Now, the Yahoo runs this cluster and offered to academic institutions all over the world.

(ii) Pig – A platform that permits the analysis of huge amount of data sets is Pig. Pig was developed like an Apache project and composed of a high-level language for representing data analysis programs, coupled with infrastructure for evaluating these programs. Pig infrastructure's layer composed of a compiler for a high level language that generates a sequence of MapReduce processes which are executed on top of distributed infrastructures. A textual language known as Pig Latin is used to express developers data analysis programs. Pig Latin reveals a SQL-like interface and it is characterized by a familiar interface with respect to MapReduce, major expressiveness, and reduced programming effort.

(iii) Hive – An Apache initiative that offers a data warehouse infrastructure on top of Hadoop MapReduce is Hive. Hive gives tools for ad-hoc queries, data summarization and analysis of large quantities of data. On the other hand, the framework offers the same capabilities of a classical data warehouse. However, it does not show the similar performances, particularly, in case of query latency. This is the reason why it does not make a valid solution for online transaction processing. The benefit of Hive is in the ability of providing a data warehouse infrastructure in environments where there is already a Hadoop system running. The main benefit of Hive is in the ability to scale out because it depends on the Hadoop framework.

(iv) Map-Reduce-Merge – An extension to the MapReduce model is Map-Reduce-Merge, which introduces a third phase to the standard MapReduce pipeline. This third phase is called as Merge phase, which permits efficiently merging data already sorted and partitioned through map and reduce functions. The Map-Reduce-Merge framework offers an abstraction capable to represent

the common relational algebra operators and several join algorithms and facilitates the heterogeneous related datasets management.

(v) Twister – An another extension to the MapReduce is Twister, which enables the creation of iterative executions of MapReduce processes. The model proposed by Twister gives the following extension –

(a) Configure map
(b) Configure reduce
(c) Repeats till condition true
  (1) Execute MapReduce
  (2) Apply combine operation to result
  (3) Modify condition
(d) Close

Additional features provided by Twister are the introduction of an additional phase known as combine execute at the end of the MapReduce job, the ability for map and reduce tasks to refer to static and in memory data, and other tools for data management.

Q.19. What are the various alternatives to MapReduce ? Explain.

Ans. Some alternatives to MapReduce are as follows –

(i) DryadLINQ – A Microsoft research project investigating programming model for writing parallel and distributed programs to scale from a small cluster to a large data center is Dryad. Dryad aims at offering infrastructure for automatically parallelizing the execution of application. Developers in Dryad can represent distributed applications like a collection of sequential programs which are joined together using channels. Direct Acyclic Graph (DAG) is used to represent Dryad computation. Here, in DAG, sequential programs denote nodes and the channels connecting such programs denote vertices. Due to this structure, a superset of the MapReduce model is considered by Dryad because its general application model permits expressing graphs representing MapReduce computation as well. Dryad revealed an interesting feature – dynamic modification of graph to certain extent and of dividing the execution of graph into stages, if possible. This infrastructure is employed to serve various applications and tools for parallel programming. DryadLINQ is one of them. Itis a programming environment which generates Dryad computations from the Language Integrated Query (LINQ) extensions to C#. The resulting framework ables to represent various distributed computing models and gives a solution completely integrated into the .NET framework.

(ii) Sphere – The distributed processing engine that leverages the Sector Distributed File System (SDFS) is Sphere. Sphere enables the developer to represent the computation in terms of User Defined Functions (UDFs) and

implements the stream processing model (single program multiple data). A certain combination of UDFs, which are executed against the distributed infrastructure, permits Sphere to represent MapReduce computations. For accessing data, Sphere is built on top of the sector's API. Programs that read and write streams are used to represent UDFs. A data structure which gives access to a collection of data segments is a stream. The collective execution of UDFs is obtained using the distributed execution of Sphere Process Engines (SPEs). The execution model is a master slave model controlled by client. Here, a request is sent by a Sphere client to the master node which returns available slaves. Then, the client will select the slaves where to run Sphere processes and manages the entire distributed execution.

(iii) All-pairs – For the optimized execution of data intensive workloads, All-pairs is an abstraction and a run time environment. All-pairs offers an abstraction in terms of the All-pairs function. This abstraction is general in many scientific computing domains and described as –

All-pairs (X : set, Y : set, F : function) → M : matrix

The model represented by the All-pairs function can be solved by the algorithm given below –

```
For each $i in X
    For each $j in Y
        Submit job F $i $j
```

Generally, this implementation is quite naive and gives worse performance. Apart from, other problems like dispatch latency, data distribution, possibility of failure, and available compute nodes are not handled specifically. These issues are handled by All-pairs model by introducing a specification for the nature of the problem and an engine that optimizes the distribution of tasks over a conventional cluster or grid infrastructure.

Q.20. Discuss development of parallel computing.

Ans. Parallel computation was appeared in the early 1960s. At that period, the transistor and core memory showed up. The processing unit became smaller and the memory became more compact and cheaper. The development of these technologies gave rise to the occurrence of parallel computer. During this period, the parallel computers are mostly shared memory multiprocessor systems in small scale which called the mainframe.

For a long time, parallel computing has been developing rapidly in the field of high performance computing and the parallel computing architecture has also been in constant changing. The platform relied by parallel computing is called parallel computer which is composed of multiple nodes. The task is decomposed to various nodes and runs in parallel on each node. The early parallel node is not the completely independent host between each other. It

more likes various modules of one host. In the 1990s, accompanying with the development of network equipment, MPI/PVM and the release of parallel programming standards, cluster architecture of parallel computer has been emerged. At this time, the node of hardware platform of parallel computing is the independent host. Modern architecture of parallel machine has symmetrical multi-processing (SMP), distributed shared memory (DSM) parallel machine, massively parallel processor (MPP) and parallel vector processor (PVP). The architecture of parallel machine are mainly used for high performance computing in specific areas. This is the foundation of cloud computing – the birth background of PC cluster of workstation. Parallel computing has been brought into people's lives by PC cluster due to the rapid development of the microprocessor performance and network bandwidth.

Q.21. Discuss about the implementation of parallel computing.

Ans. The implementation of parallel computing has two levels. First, it is the multi-core parallel computing and the multiple CPU in a single node. Although the parallel computing in single node is not the mainstream of implementation of cloud computing, multi-core is a very important aspect for a single node to improve the performance currently. Second, for cloud computing, the parallel between cluster nodes is emphasized more. At present, the nodes among cluster are usually connected by IP network. On the premise of enough bandwidth, each node is not restricted by geographical and space. So the parallel computing in cloud computing is called distributed parallel computing for many times. However, multi-CPU and multi-core are the trend of host development. So within a cluster, the parallel requirement of general level is existed such as the parallel between cluster nodes, multiprocessor within node internal and multi-core parallel. The parallel computing between nodes is equivalent to distributed parallel computing as we often said. Different from mature virtualization products, parallel computing has no mature products while only has the relatively mature tools.

The realization of parallel computing relies on the familiar with business by developers and users as well as the correct and skilled use of parallel tool. There are three steps for parallel application software development and use.

(i) On the demand analysis stage, according to business characteristics, the task is divided into multiple tasks, which can be executed in parallel as much as possible. This is the basis of the realization of parallel computing and it always be the most effective phase.

(ii) During the designing and coding phase, parallel tools are used for program design. The parallel technology used in this stage is the most key technology which is not only difficult but also not mature. Moreover, most of the technical personnel has not handle the technology applied at this stage at the moment.

(iii) Parallel deployment architecture and tools deployment applications are used in deployment stage. Through the deployment, this stage can achieve load balancing of occupied equipment. Load balancing in essence is also belonging to parallel computing. The method used in common is the load balancing device such as F5 switches, the ESB and etc. The technology involved in this stage is relatively mature which has been widely used in the field.

Q.22. Draw and explain MapReduce programming model.

Ans. A MapReduce computation workflow is illustrated in fig. 3.7. A collection of input data files is given by the user. This collection is represented in the form of a list of ⟨k1, v1⟩ pairs, and defines the map and reduce operations. These data files are entered into the distributed file system supporting MapReduce and divided in order to be the input of map tasks, if required. Then, the intermediate files are produced by map tasks. These intermediate files contain the collections of ⟨k2, list (v2)⟩ pairs and are stored into the distributed file system. The values corresponding to the similar keys might be aggregated by the MapReduce runtime. The intermediate files are then given as input to reduce tasks that eventually generate output data files in the form of list (v2). Generally, the reduce task operation is represented as an aggregation of all the values that are mapped by a particular key. The MapReduce runtime is responsible for the creation of map and reduce tasks, and how many tasks are linked to a single reduce task. The distributed file system supporting MapReduce is responsible for the way in which files are moved and stored.

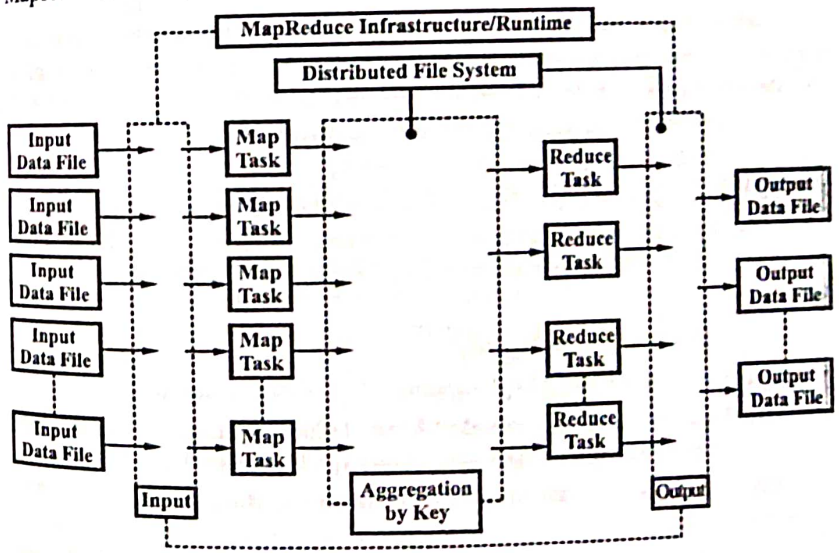


Fig. 3.7 MapReduce Computation Workflow

Q.23. Discuss in detail about Hadoop MapReduce.

Ans. A Hadoop MapReduce job mainly consists of two user-defined functions – Map and Reduce. The input of a Hadoop's MapReduce job is a set of key-value pairs (k, v) and the map function is called for each of these pairs. The map function produces zero or more intermediate key-value pairs (k', v'). Then, the Hadoop's MapReduce framework groups these intermediate key-value pairs by intermediate key k' and calls the reduce function for each group. Finally, the reduce function produces zero or more aggregated results.

The term MapReduce actually refers to two separate and distinct tasks that Hadoop programs perform they are Mappers and Reducers –

Map Job – The first is the map job, which takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs).

The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.

The map function produces zero or more intermediate key-value pairs (key', value'). Map function takes one pair of data with a type in one data domain, and returns a list of pairs in a different domain. After that, the MapReduce framework collects all pairs with the same key from all lists and groups them together, creating one group for each key.

$$Map(k1, v1) \rightarrow list(k2, v2)$$

Reduce Job – The second is the reduce job, which takes the output of a map job as input and combines those data tuples into a smaller set of tuples. The Reduce function is then applied in parallel to each group, which in turn produces a collection of values in the same domain.

$$Reduce(k2, list(v2)) \rightarrow list(v3)$$

Reduce stage is the combination of the Shuffle stage and the Reduce stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

5 Step Process of MapReduce –

Step 1 – Prepare the Map( ) Input – Set of key-value pairs (k, v)

Step 2 – Run the User-provided Map( ) Code – Generate intermediate key-value pairs (key', value') and lists $Map(k1, v1) \rightarrow list(k2, v2)$

Step 3 – "Shuffle" the Map Output to the Reduce Processors – The MapReduce system designates Reduce processors, assigns the k2 key-value each processor should work on. That is, worker nodes redistribute data based on the output keys (k2) such that all data belonging to one key is located on the same worker node.

Step 4 – Run the User-provided Reduce( ) Code – Reduce( ) is run exactly once for each k2 key value produced by the Map step.

Step 5 – Produce the Final Output – The MapReduce system collects all the Reduce output, and sorts it by k2 to produce the final outcome.

As the sequence of the name MapReduce implies, the reduce job is always performed after the map job. Below fig. 3.8 shows the MapReduce work process.

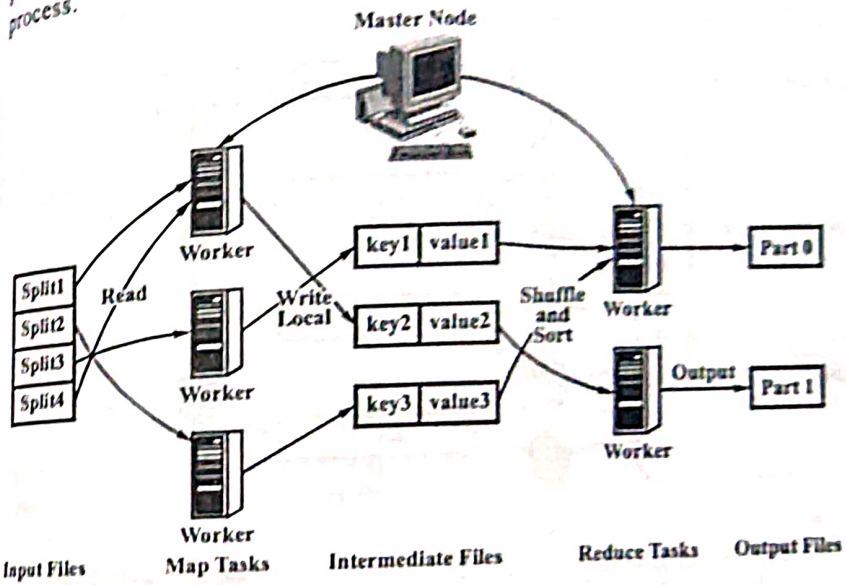


Fig. 3.8 MapReduce Working Process

Q.24. Explain the MapReduce programming model of parallel computing.

Ans. A MapReduce is a programming architecture based on cloud computing system proposed by Google and is a kind of parallel programming model which can deal with huge amounts of data for parallel computing of large-scale data set. The concept and main idea of "Map" and "Reduce" are both originated from functional programming language and vector programming language. Due to that MapReduce has common characteristics of functional and vector programming language, this programming model is particularly suited for searching, mining, analyzing and machine intelligence learning of unstructured and structured mass data. It can greatly simplify the difficulty of parallel programming. Originally, some computational problems are very simple but we have to deal with complicated details, fault-tolerant mechanisms as

well as the data distribution policy, load balance and so many details which lead to the complicated and difficult program code. Therefore, all these problems in computing applications can be wrapped in a library and be responsible by a system as a whole. Otherwise, the computing task of data is abstracted to Map and Reduce. The source of this abstraction is based on the Lisp and Map and Reduce concept of many kinds of functional language. Many computing tasks of data are related to Map. The Map is responsible for processing each logical record among input data and produces a set of intermediate Key/Value pairs, and then performs Reduce operation to all intermediate results which have the same key. The intermediate results from Map should be merged and then generated the final results.

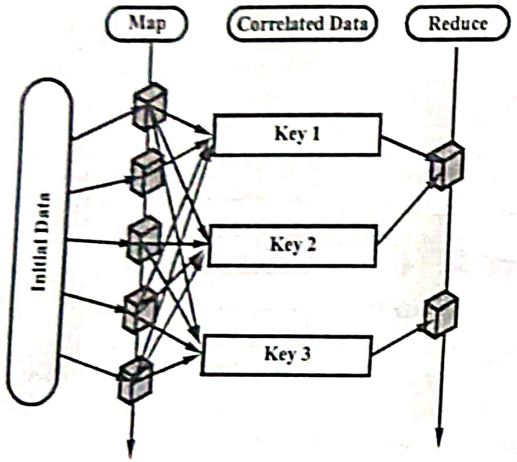


Fig. 3.9 MapReduce Programming Model of Parallel Computing

For large-scale data set operation, MapReduce is managed and controlled by a main node. The concrete computing tasks is accomplished by sending to various sub nodes from the master. In this way, MapReduce achieves reliable execution and fault-tolerant mechanisms of parallel tasks. In each time period, the master node will mark the working condition of various sub notes. Once marked for death, all the tasks of this node will be assigned to other sub nodes for re-performance.

Compared with traditional distributed programming mode, MapReduce has packaged the parallel processing, fault tolerance, localized calculation, load balance and other details and also provided a simple and powerful interface. By this interface, large data computing tasks can be concurrent and distributed automatically. During operation, system can solve the distribution problem of input data, which came across the program execution and scheduling of cluster nodes and deal with the failure of processing nodes and communications

between the management nodes. Therefore, the compilation of parallel programs becomes easier. The programmers can use the resources of large distributed systems without the concurrent processing or distributed programming experience. Meanwhile, it has great economic efficiency by high performance cluster composed by general PC to achieve super high performance. And MapReduce also has good generality.

Q.25. Describe relational operations in MapReduce.

Ans. Some relational algebra operations are as follows –

(i) Selection – Apply a condition C to each tuple in the relation and produce as output only those tuples that satisfy C. The output of this selection is denoted by $\sigma_C(R)$.

(ii) Projection – For some subset S of the attributes of the relation, produce from each tuple only the components for the attributes in S. The output of this projection is denoted by $\pi_S(R)$.

(iii) Union, Intersection and Difference – These well-known set operations apply to the sets of tuples in two relations that have the same schema. There are also bag (multiset) versions of the operations in SQL, with somewhat unintuitive definitions.

(iv) Natural Join – Given two relations, compare each pair of tuples, one from each relation. If the tuples agree on all the attributes that are common to the two schemas, then produce a tuple that has components for each of the attrtibutes in either schema and agrees with the two tuples on each attribute. If the tuples disagree on one or more shared attributes, then produce nothing from this pair of tuples. The natural join of relations R and S is denoted R ⋈ S. While we shall discuss executing only the natural join with MapReduce, all equijoins (joins where the tuple-agreement condition involves equality of attributes from the two relations that do not necessarily have the same name) can be executed in the same manner.

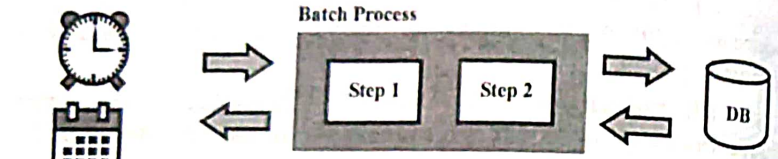
Q.26. Write short note on batch processing.

Ans. Batch processing is a store-and-process model, where data are collected, entered, processed and then the batch results are produced. There are two kinds of this model. First, when results are produced by a scheduled task it is called traditional batch processing. On the other hand, there is the service-oriented architecture or simply SOA batch processing. In this model tasks are performed when a user requires the results of an specific batch. It is widely used in relational data. Both models are shown in fig. 3.10.

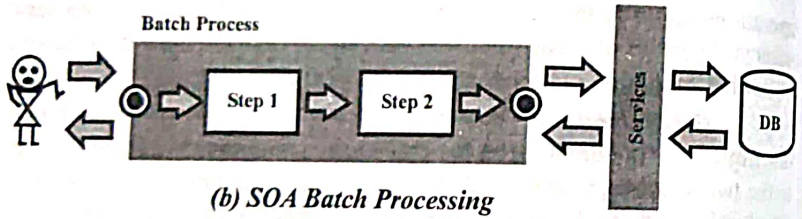
Batching processing is used in environments where it is required to process all the data at a time. However, processing all information at a time presents

disadvantages. First, it has turnaround time. This means the time taken between submitting the job and getting the output, which includes the information related to the jobs are included in the job. For this reason it is a highly time consuming model when there exists huge amount of data. All processes would produce a nonlinear behaviour and that is what produces uncertainty about the execution time.

For these reasons batch processing is commonly used in scenarios where there is a large amount of data processed and saved. Also, when a real time response is not required.



(a) Traditional Batch Processing



(b) SOA Batch Processing

Fig. 3.10 Batch Processing

Q.27. Discuss in detail about Map and Reduce operation ?

Ans. The Map operation applies computation of key/value pairs in an input and Reduce operation combines all the result value that is computed from the result of Map operation. As shown in fig. 3.11, the users divide the input files in different blocks of 128 MB size and these blocks generate the number of copies program in the clusters. Every cluster has different programs with one master node and several data nodes. Data nodes are also known as worker nodes and may be assigned Map work or Reduce work by the master node.

Once the user defines the input files, the master node assigns the worker node for Map function. Those worker nodes who are assigned for Map work reads files from different input files and writes the file in local disk. Once the Map worker nodes finished their work by writing the result in local disk, another sets of worker nodes are assigned for Reduce function. The assigned worker nodes read the files from local disk and write it to the output files. In this way, the retrieved process is completed in the Hadoop MapReduce.

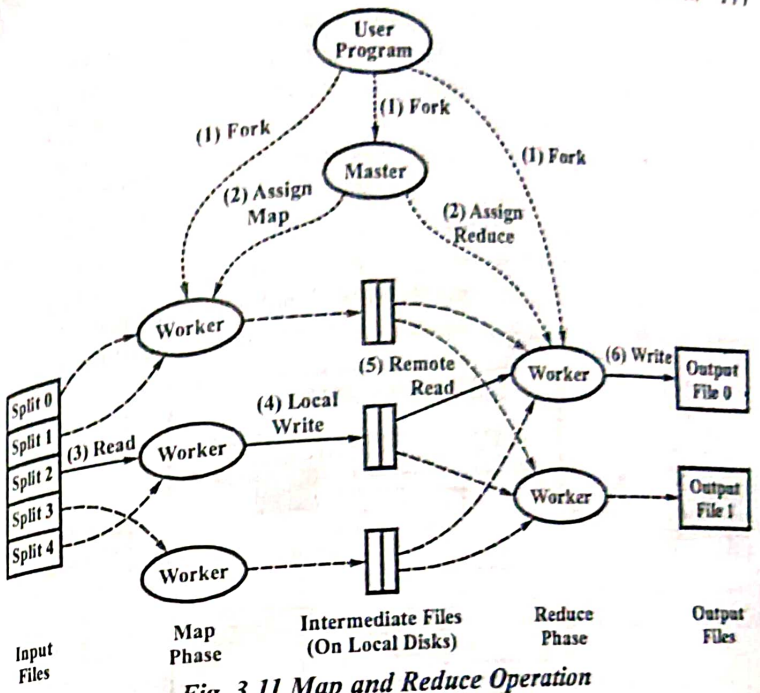


Fig. 3.11 Map and Reduce Operation

Q.28. What is Hadoop ? Explain main components of Hadoop.

Ans. Hadoop is an Apache-managed software framework created using MapReduce and Big Table. Hadoop allows applications based on MapReduce to run on large clusters of commodity hardware. The project has become the basis for the computing architecture underlying Yahoo!'s business. Hadoop is designed to parallelize data processing across computing nodes to speed computations and diminish latency. Two major components of Hadoop exist – a massively scalable distributed file system that can support petabytes of data, and a massively scalable MapReduce engine that computes results in batches.

Components of Hadoop – Two main components of Hadoop are as follows–

(i) The Hadoop Distributed File System (HDFS) – HDFS is the storage system for a cluster. When data lands in the cluster, HDFS breaks it into pieces and distribute those pieces among the different servers participating in the cluster. Each server stores just a small fragment of the complete data set and each piece of data is replicated on more than one server.

(ii) MapReduce – Because Hadoop stores the entire data set in small pieces across a number of servers, analytical jobs can be distributed in parallel to each of the servers storing part of the data. Each server evaluates the question against its local fragment simultaneously and reports its result back for collation

into a comprehensive answer. MapReduce is the agent that distributes the work and collects the results. Both HDFS and Map Reduce are designed to continue to work even if there are failures. HDFS continuously monitors the data stored on the cluster. If a server becomes unavailable, a disk drive fails or data is damaged due to hardware or software problems, HDFS automatically restores the data from one of the known good replicas stored elsewhere on the cluster. MapReduce monitors the progress of each of the servers participating in the job, when an analysis job is running. If one of them is slow in returning an answer or fails before completing its work, MapReduce automatically starts another instance of the task on another server that has a copy of the data.

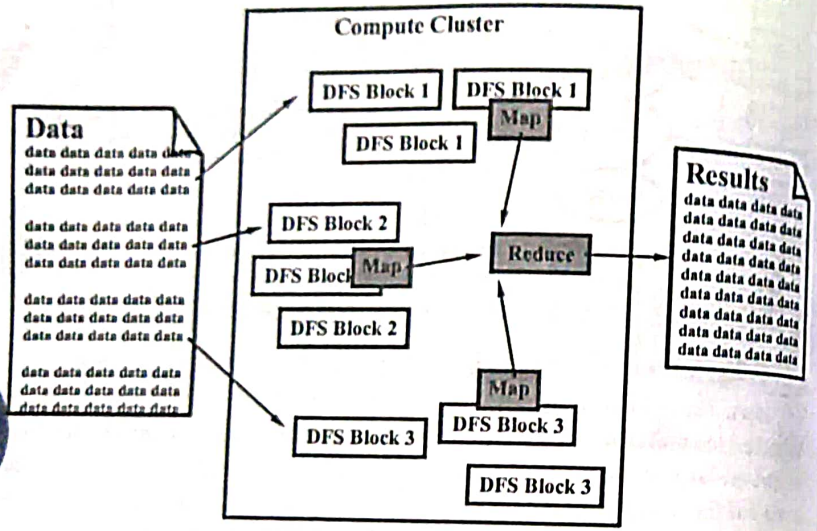


Fig. 3.12 HDFS & MapReduce

Because of the way that HDFS and MapReduce work, Hadoop provides scalable, reliable and fault-tolerant services for data storage and analysis at very low cost.

Q.29. Explain the ecosystem of Hadoop.

Ans. Hadoop is an open source framework maintained by the Apache Foundation for reliable, scalable and distributed computing. According to the website hadoop.apache.org, the components of Hadoop are defined as projects which function different to each other's. Some of the widely used Hadoop components are as follows –

(i) Pig – It is a platform for HDFS. It consists of a compiler for MapReduce programs and a high-level language called Pig Latin. It provides a way to perform data extractions, transformations and loading, and basic analysis without having to write MapReduce programs.

(ii) Hive – It is a distributed data warehouse. A data warehouse and SQL-like query language that presents data in the form of tables. Hive programming is similar to database programming. (It was initially developed by Facebook).

(iii) HBase – It is a non-relational, distributed database that runs on top of Hadoop. HBase tables can serve as input and output for MapReduce jobs.

(iv) Zookeeper – It is an application that coordinates distributed processes.

(v) Mahout – Mahout is a data mining software that can be easily scalable. Mahout offers java libraries or scalable machine learning algorithm which can be used for analyzing the data. These machine learning algorithms allow user to perform a task such as classification, clustering, association rule analysis, and predictive analysis.

(vi) Cassandra – Hadoop Cassandra provides database that can be easily scalable and highly available without interruption in the job performance.

(vii) Chukwa – Chukwa is a data collections system which is mainly used for displaying, monitoring, and analyzing the outcomes of the collected data.

(viii) Spark – Spark is a computing system which is used for configuring the Hadoop cluster for fast processing of Hadoop data. Spark does not use MapReduce job of execution engine to run the job. It uses its own distributed runtime to complete the job.

(ix) Tez – Tez is a data-flow programming language build in the Hadoop Yarn to execute an arbitrary DAG of tasks to process data for both batch and interactive use-case.

(x) Avro – Avro is used for data serialization which provides a container file for storing persistent data. Avro was created by Doug Cutting for making Hadoop to be writable in many programming languages such as C, C++, C#, Java, JavaScript, Python, Ruby.

(xi) Ambari – It is a web interface for managing, configuring and testing Hadoop services and components.

(xii) Flume – It is a software that collects, aggregates and moves large amounts of streaming data into HDFS.

(xiii) Sqoop – It is a connection and transfer mechanism that moves data between Hadoop and relational databases.

(xiv) Oozie – It is a Hadoop job scheduler.

The Hadoop ecosystem is shown in fig. 3.13.

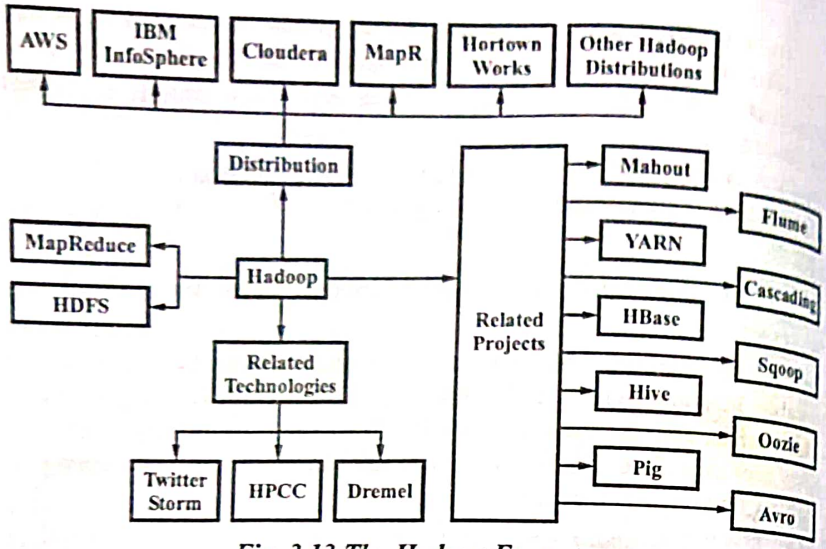


Fig. 3.13 The Hadoop Ecosystem

Q.30. Explain the application of Hadoop.

Ans. Now-a-days, with the rapid growth of the data volume, the storage and processing of Big Data has become the most pressing needs of the enterprises. Hadoop as the open source distributed computing platform has become a brilliant choice for the business. The users can develop their own distributed applications on Hadoop and processing Big Data even if they do not know the bottom-level details of the system. Due to the high performance of Hadoop, it has been widely used in many companies.

(i) Hadoop in Yahoo! – Yahoo! is the leader in Hadoop technology research and applications. It applies Hadoop on various products, which include the data analysis, content optimization, anti-spam e-mail system, and advertising optimization. Hadoop has also been fully used in user interests' prediction, searching ranking, and advertising location.

In the Yahoo! home page personalization, the real-time service system will read the data from the database to the interest mapping through the Apache. Every 5 minutes, the system will rearrange the contents based on Hadoop cluster and update the contents every 7 minutes.

Concerning spam e-mails, Yahoo! uses the Hadoop cluster to score the e-mails. Every couple of hours, the Yahoo! will improve the anti-spam e-mail model in the Hadoop clusters and the clusters will push 5 billion times of e-mails' delivery every day. At present, the largest application of the Hadoop is the Search Webmap of Yahoo!. It has been run on more than 10000 Linux cluster machines.

(ii) Hadoop in Facebook – It is known that Facebook is the largest social network in the world. From 2004 to 2009, Facebook has over 800 million active users. The data created everyday is huge. This means that Facebook is facing the problem with big data processing which contains content maintainance, photos sharing, comments, and users access histories. These data are not easy to process so Facebook has adopted the Hadoop and Hbase to handle it.

Q.31. What are the advantages of Hadoop ? Explain Hadoop architecture with proper diagram. ***(R.G.P.V., May 2019)***

Ans. **Advantages of Hadoop –**

(i) The scalability and elasticity of free open source Hadoop running on standard hardware allow organizations to hold onto more data and take advantage of all their data to increase operational efficiency and gain competitive edge. Hadoop supports complex analyses across large collections of data at one tenth the cost of traditional solutions.

(ii) Hadoop handles a variety of workloads, including search, log processing, recommendations systems, data warehousing and video/image analysis.

(iii) Apache Hadoop is an open-source project by the Apache Software foundations. The software was originally developed by the world's largest Internet companies to capture and analyze the data that they generate. Unlike traditional, structured platforms Hadoop is able to store any kind of data in its native format and to perform a wide variety of analyses and transformation on that data. Hadoop stores terabytes and even petabytes of data inexpensively. It is robust and reliable and handles hardware and system failures automatically without losing data analyses.

(iv) Hadoop runs on clusters of commodity servers and each of those servers has local CPUs and disk storage that can be leveraged by the system.

Hadoop Architecture – Hadoop is an open-source framework that allows users to store and process big data in a distributed environment across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines with high degree of fault tolerance. Data in a Hadoop cluster is broken down into smaller pieces and

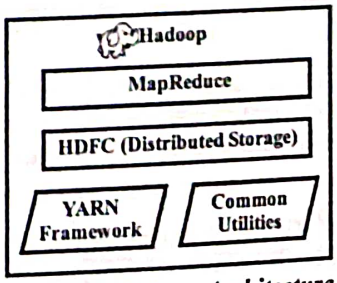


Fig. 3.14 Hadoop Architecture

distributed throughout the cluster like the Map and Reduce functions that are executed on smaller subsets of larger data sets, and this provides the scalability needed for big data processing.

Hadoop framework includes four models –

(i) Hadoop Common – They contain Java libraries and utilities that are required by other Hadoop modules. The Java libraries provide file system and OS level abstraction. It contains necessary Java files and scripts that are required to start Hadoop.

(ii) Hadoop Yarn – YARN is a cluster management technology. It is one of the key features in second-generation of Hadoop, designed from the experience gained from the first generation of Hadoop. YARN provides resource management and a central platform to deliver consistent operations, security and data governance tools across Hadoop clusters.

(iii) HDFS (Hadoop Distributed File System) – It is a distributed file system that provides high throughput computing access to application data.

(iv) Hadoop MapReduce – For large scale data processing this is programming model.

Q.32. Write short note on employing Hadoop MapReduce. Also describe it's features and applications.

Ans. A distributed data processing framework called MapReduce. In other words, MapReduce is a framework for processing parallelizable problems access large datasets using a large number of computers, collectively referred to as a cluster or a grid. Processing can occur on data stored either in a filesystem or in a database (unstructured & structured).

The features of Hadoop MapReduce are as follows –

(i) The programming model is simple yet expressive. A large number of tasks can be expressed as MapReduce jobs. The model is independent of the underlying storage system and is able to process both structured and unstructured data.

(ii) It achieves scalability through block-level scheduling. The run-time system automatically splits the input data into even-sized blocks and dynamically schedules the data blocks to the available nodes for processing.

(iii) It offers fault tolerance whereby only tasks on failed nodes have to be restarted.

The applications of MapReduce are as follows –

(i) Large scale machine learning problems

(ii) Clustering problems for Google News
(iii) Extracting data for reports of popular queries
(iv) Extracting properties of Web pages for various purposes
(v) Processing satellite image data
(vi) Language model processing for statistical machine translation
(vii) Large-scale graph computations
(viii) Index building for various search operations
(ix) Spam detection
(x) Various data mining applications.

Q.33. Explain the overview of MapReduce execution in Hadoop with the help of example.

Ans. The map tasks are distributed across multiple machines by automatically partitioning the input data into a set of M splits. These splits can be processed in parallel by different machines. Reduce tasks are distributed by partitioning the intermediate key space into R pieces using a partitioning function (e.g. hash (key) mod R). The number of partitions (R) and the partitioning function are specified by the user.

When the user program calls the MapReduce( ) function, the following occurs –

(i) The MapReduce library splits the input files into M pieces (usually 16-64 MB per piece) and starts up many copies of the program on a cluster of machines.

(ii) One of the copies of the program is the master as previously specified. The rest are workers that are assigned work by the master. There are M map tasks and R reduce tasks to assign. The master picks the idle workers and assigns each one either a map or a reduce task.

(iii) A worker assigned with a map task reads the corresponding input split. It parses key/value pairs out of the input data and passes each pair to the user-defined map function. The intermediate key/value pairs produced by the function are buffered in memory.

(iv) Periodically, these buffered pairs are written to the local disk and partitioned into R regions by the partitioning function. The locations of these pairs are passed back to the master who is responsible for forwarding these locations to the reduce workers.

(v) When a reduce worker is notified about these locations, it uses remote procedure calls (RPCs) to read the buffered data from the disks of the map workers. When a reduce worker has read all intermediate data for its

partition, it sorts it by the intermediate keys to group together all occurrences of the same key. If the amount of intermediate data is too large to fit in the memory, an external sort is used.

(vi) The reduce worker iterates over the sorted intermediate data and for each unique intermediate key, it passes the key and the corresponding set of intermediate values to the user's reduce function. The output of the reduce function is appended to a final output file for this reduce partition.

(vii) When all map and reduce tasks have completed, the master wakes up the user program. At this point, the MapReduce call in the user program returns back to the user code. After successful completion, the output of the MapReduce execution is available in the R output files.

To detect failure, the master pings every worker periodically. If no response is received from a worker in a certain amount of time, the master marks the worker as failed. Any map tasks completed by the worker are reset back to their initial idle state, and therefore become eligible for scheduling on other workers. Similarly, any map task or reduce task in progress on a failed worker is also reset to idle and becomes eligible for rescheduling.

Completed map tasks are re-executed when failure occurs because their output is stored on the local disk(s) of the failed machine and is therefore inaccessible. Completed reduce tasks do not need to be re-executed since their output is stored in a global file system.

Example of a MapReduce – Assume we have five files, and each file contains two columns, a key and a value in Hadoop terms that represent a city and the corresponding temperature recorded in that city for the various measurement days. This example is made very simple so it's easy to follow. We can imagine that a real application contain millions or even billions of rows.

Delhi, 31
Mumbai, 32
Chennai, 33
Calcutta, 32
Delhi, 24
Calcutta, 34
Chennai, 38
Delhi, 27
Calcutta, 33
Chennai, 37.

Out of all the data collected, to find the maximum temperature for each city across all of the data files (note that each file might have the same city represented multiple times). Using the MapReduce framework, we can break this down into five map tasks, where each mapper works on one of the five files and the mapper task goes through the data and returns the maximum temperature for each city. The results produced from mapper task for the above data would look like this –

(Delhi, 31)
(Mumbai, 32)
(Chennai, 38)
(Calcutta, 34)

Q.34. Give uses of MapReduce.

Ans. Uses of MapReduce are as follows –

At Google –

(i) Index building for Google Search
(ii) Article clustering for Google News
(iii) Statistical machine translation.

At Yahoo! –

(i) Index building for Yahoo! Search
(ii) Spam detection for Yahoo! Mail.

At Facebook –

(i) Ad optimization
(ii) Spam detection.

Q.35. Give the limitations of MapReduce.

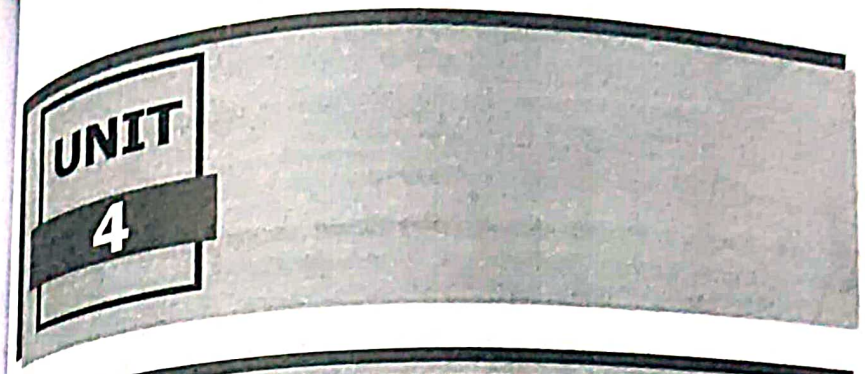
Ans. There are following four main limitations of the MapReduce –

(i) The Bottleneck of JobTracker – The JobTracker should be responsible for jobs allocation, management, and scheduling. It should also communicate with all the nodes to know the processing status. It is obvious that the JobTracker which is unique in the MapReduce, task too many tasks. If the number of clusters and the submission jobs increase rapidly, it will cause network bandwidth consumption. As a result, the JobTracker will reach bottleneck and this is the core risk of MapReduce.

(ii) The TaskTracker – Because the jobs allocation information is too simple, the TaskTracker might assign a few tasks that need more sources or need a long execution time to the same node. In this situation, it will cause node failure or slow down the processing speed.

(iii) Jobs Delay – Before the MapReduce starts to work, the TaskTracker will report its own resources and operation situation. According to the report, the JobTracker will assign the jobs and then the TaskTracker starts to run. As a consequence, the communication delay may make the JobTracker to wait too long so that the jobs cannot be completed in time.

(iv) Inflexible Framework – The MapReduce currently allows the users to define its own functions for different processing stages, the MapReduce framework still limits the programming model and the resources allocation.



CLOUD SECURITY FUNDAMENTALS, VULNERABILITY ASSESSMENT TOOL FOR CLOUD, PRIVACY AND SECURITY IN CLOUD, CLOUD COMPUTING SECURITY ARCHITECTURE, GENERAL ISSUES, TRUSTED CLOUD COMPUTING

Q.1. Define cloud security. ***[R.G.P.V., Dec. 2015 (MCA)]***

Ans. Cloud computing security refer to the set of procedures, processes and standards designed to provide information security assurance in a cloud computing environment. Cloud computing security addresses both physical and logical security issues across all the different service models of software, platform and infrastructure. It also addresses how these services are delivered.

Q.2. What are the properties that software must have to be considered secure ?

Ans. A software must have the following three properties to be considered secure –

(i) Dependability – Software that runs predictably and works properly under different conditions, including when under attack or executing on a malicious host.

(ii) Survivability (Resilience) – Software that is tolerant of or resistant to attacks and is able to recover rapidly with as minimum harm as possible.

(iii) Trustworthiness – Software that has a smaller number of vulnerabilities or no vulnerabilities or weaknesses that can harm the software's dependability.

Q.3. Write short note on audit log.

Ans. A set of records that collectively offer documentary evidence of processing, utilized to help in tracing from original transactions forward to related records and reports, and/or backward from records and reports to their component source transactions is an ***audit trial*** or ***log***.

The following should be kept in audit logs –

(i) The transaction's date and time
(ii) At which terminal the transaction was processed
(iii) Who processed the transaction
(iv) Different security events associated with the transaction.

Q.4. Give at least two reasons that cloud security system is different from traditional security system. ***[R.G.P.V., June 2015 (MCA)]***

Ans. The differences between cloud security and traditional security stems from the sharing of infrastructure on a massive scale. Users spanning different corporations and trust levels often interact with the same set of computing resources. Public cloud services are increasingly being offered by a chain of providers, all storing and processing data externally in multiple unspecified locations. The difference between traditional systems and cloud computing is the confidentiality based on encryption is hardly possible, and the inexistent user control on the physical level. This affects directly to approaches based on electronic identity cards, which now must be cloud compatible.

Q.5. What are the key mechanisms use to protect data in cloud storage? ***[R.G.P.V., June 2015 (MCA)]***

Or

Describe how the data security is handled in cloud. ***[R.G.P.V., Dec. 2017 (MCA)]***

Ans. Following are the key mechanisms for protecting data –

(i) Access control (ii) Auditing
(iii) Authentication (iv) Authorization.

The core technology for protecting data in transmit to and from the cloud as well as data stored in the cloud is encryption. The goal of encrypted cloud storage is to create a virtual private storage system that maintains confidentiality and data integrity while maintaining the benefits of cloud storage.

Q.6. Discuss different attributes of cloud security. ***[R.G.P.V., June 2016 (BE)]***

Ans. Cloud security attributes belong to broadly into the following categories –

(i) Confidentiality, Privacy and Trust – These are well known basic attributes of digital security such as authentication and authorisation of information as well as protecting privacy and trust.

(ii) Physical Protection of Enterprise Cloud Assets – This category belongs to protecting enterprise cloud centers and its assets.

(iii) Enterprise Cloud Services Security – This includes security on all its services such as SaaS, PaaS and IaaS. This is the key area of attention needed for achieving enterprise cloud security.

(iv) Data Security – This category is again paramount for sustaining enterprise cloud technology. This includes protecting and recovering planning for enterprise cloud data and service centers. It is also important to secure data in transactions.

Q.7. Explain different cloud security services. ***[R.G.P.V., Dec. 2013 (MCA)]***

Or

Discuss the different cloud security services. ***[R.G.P.V., Dec. 2014 (BE), 2015 (BE)]***

Or

Explain cloud security services. ***[R.G.P.V., Dec. 2015 (MCA)]***

Or

Explain the categories of security services provided for information over the cloud. ***[R.G.P.V., June 2016 (BE)]***

Ans. The different cloud security services are as follows –

(i) Authentication – Authentication refers to the testing or reconciliation of evidence of a user's identity. It creates the user's identity and makes sure that users are who they claim to be. Consider, for example, that a user provides an identity to a computer login screen and then has to give a password. The computer system authenticates the user by verifying that password belongs to the same user providing the ID.

(ii) Authorization – Authorization means the rights and privileges granted to a user that provide access to computer resources and information assets.

(iii) Accountability – Accountability means the ability to determine the actions and behaviours of an individual within a cloud system and to recognize that specific individual. Audit trails and logs help accountability. They can also be used to do postmortem studies in order to analyze historical events and the individuals related with those events.

(iv) Auditing – A one-time or periodic event to evaluate security is a system audit.

Information technology (IT) auditors are of two types – internal and external. Internal auditors work for a given organization, while external auditors do not. External auditors are certified public accountants or other audit professionals that do an independent audit of an organization's financial statements. Internal auditors have a much broader mandate compared to external auditors, like checking for compliance and standards of due care, auditing operational cost efficiencies, and recommending the suitable controls.

The following functions are audited by IT auditors –

(a) System development standards

(b) System and transaction controls
(c) Backup controls
(d) Data center security
(e) Data library procedures
(f) Contingency plan.

Besides, IT auditors may suggest enhancements to control, and take part in a system's development process to support an organization avoid expensive reengineering after the system's implementation.

Q.8. Write a brief note on cloud security design principles.
[R.G.P.V., June 2015 (BE)]

Or

Discuss the various cloud security design principles.
[R.G.P.V., May 2018 (MCA)]

Ans. The various cloud security design principles are as follows –

(i) Least Privilege – This principle requires that an individual, process, or other type of entity should be provided the minimum privileges and resources for the minimum time needed to finish a task. This principle decreases the opportunity for unauthorized access to important information.

(ii) Separation of Duties – This principle needs that completion of a particular sensitive activity or access to sensitive objects relies on the satisfaction of a plurality conditions. Consider, for example, an authorization would need signatures of two or more individual, or the arming of a weapons system would need two individuals with distinct keys. Therefore, in order to compromise the system, separation of duties forces collusion among entities.

(iii) Defense in Depth – This is the application of multiple layers of protection wherein a subsequent layer will offer protection if a previous layer is broken.

(iv) Fail Safe – This refers to that when a cloud system fails it should fail to a state where the security of the system and its data are not compromised.

(v) Economy of Mechanism – This principle promotes easy and comprehensible design and implementation of protection mechanisms, so that unwanted access paths can be identified and removed or do not exist.

(vi) Complete Mediation – In this principle, each request by a subject to access an object in a computer system follows an effective and valid authorization procedure. The following are included in the complete mediation –

(a) Identification of the entity requesting for the access
(b) Verification of the request that it has not altered since its initiation

(c) Application of the suitable authorization procedures
(d) The same entity reexamines the previously authorized requests.

(vii) Open Design – Some think that the encryption algorithm should be kept secret to be more hard to break. In contrast, others feel that exposing the algorithm to review and study by experts at large while keeping the encryption key secret results in a stronger algorithm since the experts have a higher probability to find weaknesses in it. Generally, the latter approach is more effective, except in the case of organizations like the National Security Agency (NSA), which uses the best cryptographers and mathematicians.

Mostly, an open-access cloud system design offers a more secure authentication method. Security of such mechanisms relies on protecting passwords or keys.

(viii) Least Common Mechanism – According to this principle, a minimum number of protection mechanisms should be common to multiple users, because shared access paths can be sources of unauthorized information exchange. The least common mechanism enhances the least possible sharing of common security mechanisms.

(ix) Psychological Acceptability – It means the easy to use and intuitiveness of the user interface that controls and interacts with the cloud access control mechanisms.

(x) Weakest Link – The security of a cloud system is as good as its weakest component. Therefore, it is necessary to recognize the weakest mechanisms in the security chain and layers of defense, and enhance them so that risks to the system are alleviated to an acceptable level.

(xi) Using Existing Components – In many cases, a cloud implementation security mechanisms might not be used to their maximum capability or configured properly. The security posture of an information system will be improved by reviewing the state and settings of the extant security mechanisms and ensuring that they are working at their optimum design points.

One other approach to enhance cloud system security by using existing components is to divide the system into defended subunits. Now, if a security mechanism is used in one sub-unit, it will not affect the other sub-units. This will result in minimum damage to the computing resources.

Q.9. What are the different secure cloud software requirements ?

Or

Explain the requirements of secure cloud software.
[R.G.P.V., Dec. 2016 (BE)]

Ans. The following three security needs are shared by all software –

(i) It must rely on expected operating conditions, and remain dependable under hostile operating conditions.

(ii) It must be reliable in its own behaviour, and in its inability to be compromised by an attacker via exploitation of vulnerabilities or insertion of malicious code.

(iii) It must be resilient enough to recover rapidly to full operational capability with a minimum of damage to itself, the resources and data it handles, and the external components with which it interacts.

Q.10. Explain the cloud information security fundamentals introduced in cloud security management. [R.G.P.V., Dec. 2014 (BE), June 2015 (BE)]

Ans. The three primary goals of information security are confidentiality, integrity and availability.

(i) Confidentiality – Confidentiality assures you that data cannot be viewed by unauthorized people. Confidentiality is concerned with preventing the unauthorized disclosure of sensitive information. The disclosure could be intentional or it could be unintentional.

(ii) Integrity – Integrity assures you that data has not been changed without your knowledge. The data is internally and externally consistent.

(iii) Availability – Availability assures you the reliable and timely access to cloud data or cloud computing resources by the appropriate personnel. Availability guarantees that the systems are functioning properly when needed.

Also refer to Q.2, Q.7, Q.8 and Q.9.

Q.11. Explain the various security benefits on the cloud.
[R.G.P.V., Nov. 2018 (MCA)]

Ans. Security benefits of using cloud are –

(i) Data Encryption – Robust data encryptions within cloud-based security systems have substantially reduced the possibilities of data breaches; these solutions offer a layered approach that consists of security intelligence, key management, and secure access controls. Cloud-based systems give the required freedom to companies to choose their users who will be accessing the data that has been outsourced to the cloud. This way, any attempts to tamper with personal or profession data can be thwarted.

Most companies face the threat of internal data theft by their employees, and stronger access controls can nip these threats in the bud. The multi-layered security features weed out the possibilities of a breach of data to a great extent. Data, irrespective of its type, needs to be protected at all times. Any violations can be hazardous to the goodwill and the functioning of an enterprise.

(ii) Avoid DDoS Attacks – Distributed Denial of Service (DDoS) attacks can result in hefty losses for entertainment companies. Hackers target the website by directing traffic from several sources to the end website, and as a result, the system gets overwhelmed. These DDoS attacks may tarnish the image of the company, as clients begin to lose trust.

Cloud-based security systems guard this imminent threat with real-time scanning of potential risks; this function is further used as a warning tool for various systems which allows for the tracking of incoming threats and attacks instantly – this enables website admins to divert the traffic to several different locations.

(iii) Regulatory Compliance – Cloud computing security solutions usually provide reliable SOC1 and SOC2 certifications to the entertainment businesses. These certifications ensure periodic scrutiny of data and all types of possible glitches. Cloud-based solutions manage the requisite infrastructure for regulatory compliance and the protection of data. Detailed AWS reports about management of security controls ensure all organizations focus on their business operations, without worrying about compliance requirements.

(iv) Secure Storage – Traditional storage solutions do not provide any protection against possible disasters that have the potential to erase required data from devices. Cloud computing allows the users to store their data safely, thereby negating any mishaps that may affect the equipment.

Cloud storage solutions offer private, public, and hybrid solutions which the businesses may choose as per their requirements. The hybrid cloud storage systems allow the users to keep their data secure in the most effective manner.

(v) Patch Management – The vulnerabilities of a website are often exploited by hackers to breach the security system of a company. Cloud service providers keep their sites up to date; further on, they ensure that no vulnerabilities exist. Moreover, cloud solutions offer real-time assistance to clients by providing companies with the option to scale cloud solutions during high traffic situations. This flexibility allows companies to reduce their cost of services substantially.

These large number of security features are quite flexible, agile, and affordable. Enhanced security features offer sufficient protection to the private and financial data of both media and entertainment companies and help to thwart data and intellectual property breaches. In this era of digitalization, where cybercrime has emerged as a norm, cloud-based solutions seem to be the best alternative to traditional security systems.

Q.12. What is a vulnerability scanner ?

Ans. A vulnerability scanner is a tool that can scan an entire system for known vulnerabilities. It typically works in conjunction with a database full of known vulnerabilities and cross checks the database with any exploits the scanner may find. The security implications behind using a scanner on a

system are –

(i) They provide reports on vulnerabilities found.

(ii) Can be automated to scan an entire system.

One of the downsides to automated reports is that sometimes it reports false positives, also known as a false alarm. If there are many false alarms, the reliability of the vulnerability scanner diminishes, and security administrators must spend many hours checking each false positive.

Q.13. Discuss about the vulnerability assessment scanners.

Ans. Vulnerability assessment scanners are tools that scan networks and provide a diagnostic report on discovered vulnerabilities. The scanner cross checks with a database of known common security holes and tries to exploit them. Any detected security holes are classified, and mentioned to the administrator along with a set of recommendations that they should take to improve the security of their network. As cloud computing demands have increased, security implications are now more vital than ever. New tools such as scanners are used to take an automated analytical approach that can determine that vulnerabilities not only exist, but also can effectively correct security flaws. A few tools some online companies using are Nessus, OpenVAS, Nexpose, or other vulnerability scanning tools. These scanners over the years have developed from unreliable tools that reported false positives, lacked scalability, and other bugs that prevented the tools from effectively scanning the network for vulnerabilities. For example, they did not provide detailed information or control services. These major vulnerability assessment tools, despite their flaws, have provided an impetus for the development of new tools.

Q.14. What are the key privacy concerns in the cloud ?

[R.G.P.V., June 2015 (MCA)]

Ans. Privacy advocates have raised many concerns about cloud computing. These concerns typically mix security and privacy. Some additional considerations to be aware of are –

(i) Storage – When the data is stored in the cloud, such a transfer may occur without the knowledge of the organization, resulting in a potential violation of the law.

(ii) Access – Data subject have a right to know what personal information is held and, in some cases, can make a request to stop processing it. This is important with regard to marketing activities. In the cloud, the main concern is the organization's ability to provide the individual with access to all personal information, and to comply with stated requests. If a data subject exercises this right to ask the organization to delete his data, will be possible to ensure that all of his information has been deleted in the cloud ?

(iii) Compliance – Clouds can cross multiple jurisdictions, for example, data may be stored in multiple countries, or in multiple states within the United States. What is the relevant jurisdiction that govern an entity's data in cloud and how is it determined ?

(iv) Retention – How long is personal information (that is transferred to the cloud) retained ? Who enforces the retention policy in the cloud, and how are expectations to this policy managed ?

Q.15. Discuss the various types of security policies.

Ans. In the corporate world, when we refer to specific policies, rather than a group policy, we generally mean those policies that are distinct from the standards, procedures, and guidelines. Policies are considered the first and highest level of documentation for strategic reasons, from which the lower-level elements of standards procedures, and guidelines flow. The various security policy types are as follows –

(i) Senior Management Statement of Policy – This is the first policy of any policy creation process. This high level policy acknowledges the importance of the computing resources to the business model.

(ii) Regulatory Policies – These policies are implemented by an organization due to compliance regulation, or other legal requirements. These policies are very detailed and specific to the industry where the organization works. These organizations may be financial institutions, public utilities, or some other kind of organization working in the public interest.

(iii) Advisory Policies – These policies are not mandatory but strongly recommended, perhaps with serious results defined for failure to follow them. An organization following such policies needs most employees to consider these policies mandatory. Most policies belong to this category.

(iv) Informative Policies – These policies exist to inform the reader. These are not implied or specified requirements. The audience for this information can be some internal or external parties.

Q.16. What are the cloud security services ? Explain different polices made to provide secure cloud computing environment.

[R.G.P.V., May 2019 (MCA)]

Ans. Cloud security service is a Web based identity and access management solution. Cloud security service allows software-as-a-service (SaaS), Platform-as-a-service (PaaS) and infrastructure-as-a-service (IaaS) providers to offer their enterprise customers the ability to deploy their existing identity infrastructure in a cloud.

Cloud security service is the cloud security broker, a collection of cloud elements that work together to provide a secure place for cloud workloads

and cloud storage. SaaS and PaaS platforms access the security broker via identity and event connectors, while the enterprise accesses the broker via an on premise secure bridge run from the data center. This secure bridge, which is firewall friendly, provides a protocol proxy, policy agent, audit agent, secure communication manager and key agent. The broker ensures that sensitive information always remains behind the firewall.

For example, IDaaS (Identity-as-a-services) is one of the cloud security services which is described as a combination of administration and account provisioning, authentication and authorization and reporting functions.

Refer to Q.15.

Q.17. Explain the term policy implementation.

[R.G.P.V., Dec. 2015 (MCA)]

Ans. A policy is one of those terms that can mean several things. For example, there are security policies on firewalls, which refer to the access control and routing list information. Standards, procedures, and guidelines are also referred to as policies in the larger sense of a global information security policy. A good, well written policy is more than an exercise created on white paper – it is an essential and fundamental element of sound security practice. A policy, for example, can literally be a lifesaver during a disaster, or it might be a requirement of a governmental or regulatory function. A policy can also provide protection from liability due to an employee's actions, or it can control access to trade secrets. Security policies and their relation hierarchically is shown in fig. 4.1.

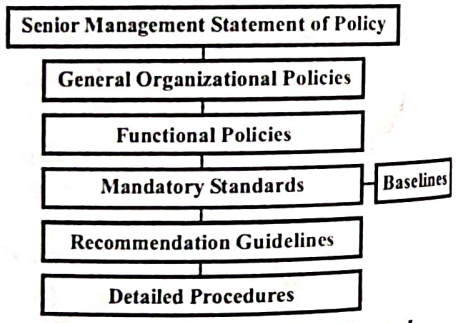


Fig. 4.1 Security Policy Hierarchy

Refer to Q.15.

Q.18. Discuss different areas of cloud policy implementation.

[R.G.P.V., Dec. 2013 (MCA)]

Ans. Following are the key areas of cloud policy implementation –

(i) Authentication and Access Control – One of the key cloud security areas is access control and is a good example to demonstrate the shared responsibility concept. PaaS and SaaS providers, for instance, can provide authentication for cloud application developers and users. On the other hand, opportunities exist for cloud subscribers to take ownership of authentication and access control to cloud for tighter integration with their identity and access management systems. Client-side access control is an integral component of their cloud security strategy for IaaS subscribers.

(ii) Consistency – An overarching and consistent policy framework is critical for successful cloud security implementation. For example, an excellent design to achieve reliable and dynamic logical separation is to apply zone-based and policy-driven security enforcement. A zone is a group of attributes they may include traditional networking parameters such as IP addresses, network protocols and port numbers. The zone may also contain information such as virtual machine (VM) and custom attributes. Approaches such as this help ensure policy consistency in a dynamic cloud environment where VMs typically move around.

(iii) Architecture – The cloud computing architecture generally includes the underlying infrastructure, various service components, and certain pervasive functions such as security and resiliency. Furthermore, cloud security has its own architectural structure.

(iv) Automation – A core tenet of the cloud computing business model is pay-per-use, meaning that elasticity is not only reflected in the infrastructure and computing power, but also in the cost structure.

(v) Governance – Cloud computing represents a dramatic shift to new technologies and new business computing models. Providers and subscribers need to ensure that their organizational governance is up to date to support these changes. From a technology perspective, cloud governance necessitates an increase in visibility and auditing capabilities.

(vi) Logical Separation – A key cloud computing benefit is its elastic computing capabilities, meaning that computing power can be ramped up or dialed down rapidly based on demand. To support such a dynamic business computing model, security should be provisioned in a similar manner. Static and physically oriented security configurations such as VLAN-based security are labor intensive and can hardly keep up with the fast pace. New approaches are needed to achieve logical separation to secure dynamic and shared environments such as multi-tenancy.

(vii) Scalability and Performance – Scalability and performance are requirements for cloud security because of the potentially massive workloads and stringent security requirements involved. Innovative technologies that can help boost performance while maintaining a high security standard is critical to cloud security implementation.

Q.19. Explain the following –

(i) Security governance (ii) Security monitoring

(iii) Risk management.

[R.G.P.V., Nov. 2018 (BE)]

Ans. (i) Security Governance – A security steering committee should be developed whose objective is to focus on providing guidance about security initiatives and alignment with business and IT strategies. This committee must clearly define the roles and responsibilities of the security team and other groups involved in performing information security functions.

Cloud security governance refers to the management model that facilitates effective and efficient security management and operations in the cloud environment so that an enterprise's business targets are achieved. This model incorporates a hierarchy of executive mandates, performance expectations, operational practices, structures, and metrics that, when implemented, result in the optimization of business value for an enterprise. Cloud security governance helps answer leadership questions such as –

(a) Are our security investments yielding the desired returns ?

(b) Do we know our security risks and their business impact ?

(c) Are we progressively reducing security risks to acceptable levels ?

(d) Have we established a security-conscious culture within the enterprise ?

Strategic alignment, value delivery, risk mitigation, effective use of resources, and performance measurement are key objectives of any IT-related governance model, security included. To successfully pursue and achieve these objectives, it is important to understand the operational culture and business and customer profiles of an enterprise, so that an effective security governance model can be customized for the enterprise.

(ii) Security Monitoring – Security management prevents unauthorized access and configuration of storage infrastructure components. For example, while deploying an application or a server, the security management task include managing user accounts and access policies, that authorizes users to perform role-based activities security monitoring in the cloud should be integrated with existing enterprise security monitoring tools using an API.

(iii) Risk Management – Risk management entails identification of technology assets; identification of data and its links to business processes, applications, and data stores; and assignment of ownership and custodial responsibilities. Actions should also include maintaining a repository of information assets. Owners have authority and accountability for information assets including protection requirements, and custodians implement confidentiality, integrity, availability, and privacy controls.

Q.20. Why cloud computing brings new threats ?

[R.G.P.V., June 2017 (MCA)]

Ans. The cloud computing brings new threats due to –

(i) Loss of control

(ii) Lack of trust

(iii) Multi-tenancy

(iv) User identity management is controlled by the cloud

(v) Consumer depends on provider to ensure

(a) Data security and privacy

(b) Resource availability

(c) Monitoring and repairing of services or resources.

(vi) User access control rules, security policies and enforcement are managed by the cloud service provider.

Q.21. What is multitenancy issue in cloud computing ?

[R.G.P.V., Dec. 2016 (BE)]

Or

Explain risk from multitenancy with respect to various cloud environment. ***[R.G.P.V., June 2017 (MCA)]***

Ans. Multitenancy is a one to many model which allows sharing of resources and costs across multiple users. The concept of multitenancy is a critical issue in cloud computing because it is directly related to security and QoS in the aspect of companies and individual. Although, it still faces a big challenge of security and privacy problem, secured multitenancy should be applied in cloud computing environments to reduce cost correlated with building computing resources, especially storage resource and to effectively manage infrastructure.

Q.22. What do you understand by security awareness ? What are its benefits ? How can we improve the security within an organization ?

Ans. Security awareness of an organization means the degree to which its personnel are collectively aware of the importance of security and security controls.

Personnel are viewed as "security aware" when they understand the requirement for security, how the viability and the bottom line are affected by security, and the daily risks to cloud computing resources.

Security awareness programs have the following benefits –

(i) They aid to prevent the fraud, waste, and abuse of computing resources.

(ii) They can enhance the effectiveness of the protection controls.

(iii) They can decrease the unauthorized actions performed by personnel.

The following activities are used to enhance security within an organization –

(i) Live/Interactive Presentations – Videos, lectures, and computer based training

(ii) Incentives – Awards, etc.

(iii) Publishing/Distribution – Bulletins, posters, and the intranet.

(iv) Reminders – Log-in banner messages and marketing paraphernalia like mugs, pens and mouse pads.

Q.23. Explain the cloud computing security architecture using suitable block diagram. ***[R.G.P.V., Dec. 2015 (BE)]***

Or

Explain cloud computing security architecture.

[R.G.P.V., Dec. 2013 (MCA), 2014 (MCA), 2014 (BE), 2015 (MCA), 2016 (BE)]

Ans. Cloud application developers have been successfully developing applications for IaaS and PaaS platforms. These platforms offter basic security features but security concerns continue to be the number one barrier for enterprise cloud adoption. Cloud security concerns range from securely configuring virtual machines deployed on an IaaS platform to managing user privileges in a PaaS cloud. The cloud services can be delivered in many flavors, i.e., in any combination of service delivery models SaaS, PaaS and IaaS (SPI), and operational models, public, private and hybrid, the cloud security concerns and solutions are context dependent. Hence the solution architecture should match these concerns and build security safeguards (controls) into the cloud application architecture.

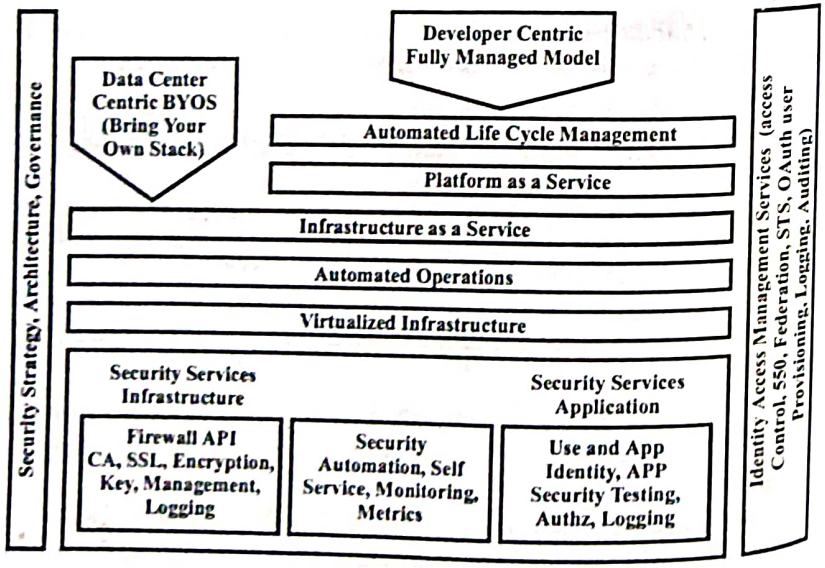


Fig. 4.2 Cloud Security Architecture Plan

As a first step, architects need to understand what security capabilities are offered by cloud platforms. The architecture for building security into cloud services is shown in fig. 4.2. Security capabilities and offerings continue to evolve and vary between cloud providers. Hence you will often discover that security mechanisms such as key management and data encryption will not be available. For example, the need for a AES 128 bit encryption service for encrypting security artifacts and keys escrowed to a key management service. For such critical services, one will continue to rely on internal security services. A "Hybrid Cloud" deployment architecture pattern may be the only viable option for such applications that dependent on internal services.

Q.24. With the help of a neat diagram explain the principal components of cloud computing security architecture. ***[R.G.P.V., June 2017 (BE)]***

Ans. Refer to Q.23.

User Layer Components –

(i) Cloud applications (ii) Programming
(iii) Tools (iv) Environments.

Service Provider Layer Components –

(i) SLA monitor (ii) Metering
(iii) Accounting (iv) Resource provisioning
(v) Scheduler and dispatcher (vi) Load balancer
(vii) Advance resource reservation monitor
(viii) Policy management.

Virtual Machine Layer Components –

(i) Virtual machines (ii) Operating systems
(iii) Monitoring of operating system.

Data Center Layer Components –

(i) Servers (ii) CPU's (iii) Memory (iv) Storage.

Q.25. Discuss the compliance issues in cloud security architecture.

Ans. The provider does not usually notify the clients of the storage location of their data in a public cloud environment. In fact, one of the cloud's fundamental characteristics is the distribution of processing and data storage. However, to consider the client's data location needs, the cloud provider should help. Besides, the cloud vendor should offer transparency to the client by providing information about used storage, processing characteristics, and other related account information.

The accessibility of a client's data by the provider's system engineers and some other employees is another compliance issue. This factor is an essential part of offering and maintaining cloud services but the act to obtain

important information should be monitored, controlled, and protected by safeguards like separation of duties. The ability of local law enforcement agencies to access a client's sensitive data is a concern in cases where information is stored in a foreign jurisdiction. Consider, for example, the situation when a government entity does a computer forensics investigation of a cloud provider under suspicion of illegal activity.

For data protection and compliance, the cloud provider's claims must be backed up by certifications, auditing, and logging. Specifically, a cloud provider should undergo a Statement on Auditing Standard #70(SAS 70) "Service Organizations" Type II Audit at a minimum. A service organization's internal controls are evaluated by this audit to ascertain whether accepted best practices are being used to protect client information. It is needed by client vendors to undergo subsequent audits to keep their SAS 70 Type II Audit certification.

The management policy related with data stored in the cloud is one other related issue. The compliance and privacy needs have to be considered when a client's engagement with the cloud provider is ended. In some situations, information is preserved on the basis of regulatory requirements and in other cases the provider should not keep a client's data in primary or backup storage if the client realizes it has been destroyed. When data is stored in a foreign jurisdiction, the data may be subjected to that country's privacy laws.

Q.26. Define the term architectural consideration.

Ans. A variety of factors affect the implementation and performance of cloud security architecture. There are general issues involving regulatory requirements, security management, adherence to standards, information classification and security awareness. Then there are more specific architecturally related areas, including trusted hardware and software, providing for a secure execution environment, establishing secure communications, and hardware augmentation through microarchitectures.

Q.27. Write short note on general issues in cloud security architecture.

Ans. Many topics influence and directly affect the cloud security architecture. They include such factors as security management, compliance, administrative issues, controls and security awareness. Compliance with legal regulations should be supported by the cloud security architecture. As a corollary, the cloud security policy should address classification of information, what entities can potentially access information, under what conditions the access has to be provided, the geographical jurisdiction of the stored data, and whether or not the access is appropriate. Proper controls should be determined and verified with assurance methods, and appropriate personnel awareness education should be put in place.

Q.28. What is trusted cloud computing ? What are its characteristics ? [R.G.P.V., May 2018 (MCA)]

Ans. Trusted cloud computing is considered as a computer security architecture which is designed to protect cloud systems from harmful intrusions and attacks. It ensures that computing resources will work in a particular, predictable way as desired.

Trusted Computing Characteristics – There might be running multiple processes concurrently in a cloud computational system. Each process can access specific memory locations and run a subset of the computer's instruction set. The execution and memory space allocated to each process is known as a ***protection domain***. This domain is extended to virtual memory to increase the real memory size. The goal of a protection domain is to protect programs from all unauthorized actions.

The total combination of protection mechanisms within a computer system is called a ***trusted computing base*** (TCB). TCB includes the hardware, software, and firmware to enforce the security policy of a computing system. These components must be protected from harmful processes. It must also offer for memory protection and make sure that the processes belonging to a domain do not access memory locations of another domain. The boundary that separates the TCB from the remainder of the system is called the ***security perimeter***. There must be a trusted path to access the TCB. Thus, a trusted computer system uses the necessary hardware and software assurance measures to allow its use in processing multiple levels of classified or sensitive information. This system satisfies the needs for reliability and security.

The trusted platform module (TPM) is used to store cryptographic keys that can be used to attest to the operating state of a computing platform and to confirm that the hardware and software configuration has not been modified.

Q.29. What is VM rootkit ? [R.G.P.V., June 2017 (BE)]

Ans. A rootkit is a software, which is used to perform some illegal operation. It is not harmful. It is used to hide entry of malware or worms into the computer system by making some hidden channels. Similarly, virtualization-based rootkits use rogue hypervisor to make a hidden channel to insert unauthorized code into system.

Q.30. What is honeypot ? What are the different types of honeypot ? [R.G.P.V., May 2019 (BE)]

Ans. According to the Lance Spitzner, "a honeypot is an information system resources whose value lies in unauthorized of illicit use of that resources". Honeypot is a useful tool for luring and trapping attackers, capturing information. Security is the essential element of any organization Web sites,

but though the security provided by the honeypots based on hardware setups are very expensive for small and medium scaled organization; a software based honeypot may be proven as a very effective security solution for these organizations. Among all these types of honeypot low-interaction honeypot is the mostly used honeypot, because it is easy to implement and manage. But the most secure and efficient honeypot type is high interaction honeypot. These honeypots provide security as well as generates a log about all entries in the system which is very helpful to find the intrusive activity in the system. But the honeypot must need to upgrade to new methods and attacks at some interval of time to provide security against new type to attacks. It can't be said as a solution but it is good supplement for the security system.

A honeypot can detect the behaviour of the attacker or the intrusion information to observe and record the details of the attacker and create a log of malicious entries and examines level, purpose, tools and methods used by the attacker so that evidence can be obtained and further actions can be taken.

Honeypot technology and traditional security system combined can build an active network security protection system.

Honeypots can be classified based on the level of interaction between intruder and system. These are low-interaction, high-interaction and medium-interaction honeypot.

(i) Low-interaction Honeypot – These types of honeypots have the limited extend of interaction with external system. FTP is the example of this type of honeypot. There is no operation system for attackers to interact with, but they implement targets to attract or detect attackers by using software to emulate features of a particular operating system and network services on a host operation system. Main advantage of this type of honeypot is that, it is very easy to deploy and maintain and it does not involve any complex architecture. With this advantage there is also some drawback of this system. That is, it will not respond accurately to exploits. This creates the limitation in ability to aid in discovering new vulnerabilities or new attack patterns. Low-interactive honeypots are a safer and easy way to gather info about the frequently occurred attacks and their sources.

(ii) High-interaction Honeypot – This is the most advanced honeypot. This type of honeypot have very higher level of interaction with the intrusive system. It gives more realistic experience to the attackers and gathers more information about intended attacks; this also involves very high risk of capturing of whole honeypot. High-interaction honeypot are most complex and time consuming to design and manage. High-interactive honeypots are more useful in the cases, where we want to capture the details of vulnerabilities or exploits that are not yet known to the outside world. This honeypots are

best in the case of "0-Day attacks". Ex – Honeynets – which are typically used for research purpose.

(iii) Medium-interaction Honeypot – These are also known as mixed-interactive honeypots. Medium-interaction honeypots are slightly more sophisticated than low-interaction honeypots, but are less sophisticated than high-interaction honeypots. It provides the attacker with a better illusion of the operation system so that more complex attacks can be logged and analysed. Ex : Honeytrap : it dynamically creates port listeners based on TCP connection attempts extracted from a network interface stream, which allows the handling of some unknown attacks.

SECURITY CHALLENGES, VIRTUALIZATION SECURITY MANAGEMENT, VIRTUAL THREATS, VM SECURITY RECOMMENDATIONS, VM-SPECIFIC SECURITY TECHNIQUES, SECURE EXECUTION ENVIRONMENTS AND COMMUNICATIONS IN CLOUD

Q.31. List and explain various cloud computing security challenges. ***[R.G.P.V., June 2015 (BE)]***

Or

What are the different security challenges in cloud computing ? Discuss each in brief. ***[R.G.P.V., May 2019 (BE)]***

Ans. The security challenges in cloud computing are as follows –

(i) Logical Storage Segregation and Multi-tenancy Security Issues– Users can store and deliver their data across the globe through Internet using cloud computing. The user does not control, and typically does not even know the location where the data is exactly stored. There is a possibility that user and their competitor's data can reside on the same physical storage device with logical segregation. That's why there is a chances of user's private data to be viewed by the other users. If the data and the information are not protected from other users then it is a major risk for the user to keep their information private in the cloud. In addition, the data is deployed on the cloud service provider's infrastructure on a multitenant model basis. This situation brings the security concerns like who maintains the audit records of the data ? Who owns the data ownership and control ownership ? To handle such sensitive situations, cloud service provider should ensure proper data isolation.

(ii) Identity Management Issues – The advancement of cloud computing based on numerous technical and business models signifies that cloud computing with an appropriate identity management can be considered as a superset of all the corresponding issues from these paradigms and many

more. As the traditional identity and access management is still facing so many challenges when considering it for cloud computing, it needs to be more secure. Unlike traditional identity management, simply managing users and services is not sufficient is cloud computing.

(iii) Insider Attacks – In cloud computing, one of the major security concerns is that the customer loses direct control over potentially business sensitive and confidential data. This needs more attention because the cloud service provider is outside the trusted domain of customer. The risk of malicious insider is the most dangerous security threats. This threat is intensified for customers of cloud services by the union of infrastructure, services and customers under a single controlling domain, with a huge lack of transparency in the way the cloud service provider services through its processes and procedures.

(iv) Virtualization Issues – Virtualization is a key element for cloud computing to achieve its objective. It can be achieved through a hypervisor. Virtualization of enterprise servers introduces noteworthy security concerns due to aggregation of risks. Associating multiple servers with one host removes the physical separation between servers, increasing the risk of undesirable cooperation of one application with others on the same host. At the same time, if an attacker gets the root to access the hypervisor, then it brings significant threats to the cloud computing. The attacker can gain access to all Guest's OS created on that virtualization server, if the attacker hacks the virtualization host machine.

Q.32. How is security provided to data at various stages in context of cloud ? ***[R.G.P.V., Dec. 2017 (BE)]***

Ans. Security provided to data at various stages in context of cloud are discussed below –

(i) At Infrastructure Level – If he/she gets all the rights of system admin since he/she can attack the system. System admin can install or execute all software to perform an attack with root privileges at each machine. If he gets the physical access of machine. He can perform the sophisticated attacks as cold boot attacks and tamper with the hardware.

(a) Protection at Infrastructure Level –

(1) All system admin right must not be get by any single person.

(2) Restricted access control policies, stringent security devices and surveillance technique should be use by provider to protect physical integrity of the hardware.

(3) Trusted computing group to identify and implement security measures at infrastructure stage provides a set of hardware and

software methods to allow construction of trusted platforms suggests use of remote attestation.

(ii) At Platform Level – At this level, provider maintains data integrity and availability.

(a) Integrity – When you download a file from Internet. It is require to check your file data is some as you want. It should not a malicious program and there is not any computer between you and server which sending you a different file. It is known as man-in-the-middle attack. You should ensure file integrity. The integrity technique protects you data from man-in-the-middle attack.

Integrity Goals –

(1) Protect data from man-in-the-middle attack.

(2) Preservation of the internal and external consistency.

(b) Confidentiality – Confidentiality helps the user to conform that unauthorized person like hacker and other person cannot be seen user data. Encryption is one of the most famous method to protect your data from seen by unauthorized person.

(c) Authentication – Refer to Q.7 (i).

(d) Defense from Intrusion and DoS Attack – The aim of DoS attack is to reduce the system performance or fully interrupt the system service. The attacker can use various method to achieve this goal. For instance the attacker sends too many login requests to a server using different random its continuously in quick succession due to which network can be overload.

(iii) At Application Level – The following rules are an integral part of the application development and deployment process –

(a) Regulatory Compliance – The target is defined in regulatory compliance that organization want to achieve to conform that they are taken steps to comply with relevant rules and regulations.

(b) Data Segregation – Data segregation refers to separate the user data which is held by broker from broker data.

(c) Availability – Availability means network, hardware, software, system are reliable and after any error they can recover immediately and fully. These must not be affected by denial of service attacks.

(iv) At Data Level – For protecting data from corruption and losses, we perform data protection technique at infrastructure level. We should also conform that data is encrypted during transit and at rest. We should also perform stringent security process, periodic audits. Cloud is secure across all layer is conformed by ethical hacking and vulnerability testing.

Q.33. Describe the top threats identified by cloud security alliance (CSA) of cloud computing. ***[R.G.P.V., Dec. 2013 (BE)]***

Ans. The top threats identified by cloud security alliance (CSA) of cloud computing are as follows –

(i) Shared Technology – Disk partitions, central processing unit caches, graphics processing units and other shared elements were never designed for strong compartmentalization. As a result, attackers focus on how to affect the operations of other cloud customers and gain unauthorized access to data.

(ii) Insecure APIs – Criminals continue to leverage new technologies to improve their reach, avoid detection, and improve the effectiveness of their activities.

(iii) Malicious Insiders – Risk arises because the cloud service providers do not disclose their hiring standards and policies; potential harm due to this particular form of attack is quite substantial.

(iv) Data Loss or Leakage – Data loss or leakage can have devastating effect on a business. Beyond the damage it can cause to one's brand and reputation, a loss could significantly affect employee, partner, and customer morale and trust.

(v) Account and Service Hijacking – Account and service hijacking, usually with stolen credentials, remains a top threat. With stolen credentials, attackers can often access critical areas of deployed cloud computing services, allowing them to compromise the confidentiality, integrity and availability of those services.

(vi) Unknown Risk Profile – It refers to exposure to the ignorance or underestimation of the very risks of cloud computing.

Q.34. Explain virtualization security management in cloud computing. ***[R.G.P.V., Dec. 2015 (BE), Nov. 2019 (MCA)]***

Ans. Threats to the virtualized infrastructure are evolving just as quickly, although the global adoption of virtualization is a relatively recent event. The virtual machine, virtual memory manager and hypervisor or host OS are the minimum set of components required in a virtual environment. They comprise virtual environment in a few different ways –

(i) Type 1 virtual environments are considered full virtualization environments and have virtual machine running on a hypervisor that interacts with the hardware.

(ii) Type 2 virtual environments are also considered full virtualization but work with a host OS.

(iii) Para-virtualized environments offer performance gains by eliminating some of the emulation that occurs in full virtualization environments.

(iv) Other type designations include hybrid virtual machines and hardware-assisted techniques.

These classifications are somewhat ambiguous in the IT community at large. From a security perspective, there is a more significant impact when a host OS with user applications and interfaces is running outside of a VM at a level lower than the other VMs. Because of its architecture, the type 2 environment increases the potential risk of attacks against the host OS.

The VMware infrastructure is managed by several users performing different roles. The roles assumed by administrators are the Virtualization Server Administrator Virtual Machine Administrator, and Guest Administrator. VMware infrastructure users may have different roles and responsibilities, but some functional overlap may occur.

Q.35. Describe virtual threats of the vulnerabilities in virtual environments.

Ans. Some of the vulnerabilities exposed to any malicious-minded individuals are –

(i) Shared Clipboard – This technology permits the transfer of data between VMs and the host. Thus, it offers a way to move data between malicious programs in VMs of different security realms.

(ii) Keystroke Logging – Some VM technologies permit the logging of keystrokes and screen updates to be sent across virtual terminals in virtual machine, writing to host files and allowing the monitoring of encrypted terminal connections within the VM.

(iii) Virtual Machine Monitoring from Another VM – One VM should not be able to directly access virtual disk of another VM on the host. However, intruders may use a hacker technique, called as ARP poisoning, to redirect packets going to or from the other VM for sniffing, if the VM platform uses a virtual hub or switch to connect the VMs to the host.

(iv) VM Monitoring from the Host – The host may affect the VM in following ways –

(a) Beginning, pausing, stapping, and restart VMs.

(b) Monitoring the applications executing within the VM.

(c) Configuring and monitoring resources available to the VMs, like CPU, memory, and disk.

(d) Adjusting the amount of memory, amount and number of virtual disks, number of CPUs, and number of virtual network interfaces available to a VM.

(e) Viewing, copying, and updating data stored on the VM's virtual disks.

(v) Virtual Machine Backdoors – A backdoor can permit intruders to carry out dangerous operations.

Q.36. Discuss the virtual machine (VM) security recommendations.

Ans. The virtual machine (VM) security recommendations include the following –

(i) VM Using Encrypted Communications – An encryption technologies, like SSH (secure shell), encrypted VPNs (virtual private networks), TLS (transport layer security) and HTTPS (secure HTTP) should be used to give secure communication links between the guest domain and the host domain, or from hosts to management systems. Encryption help to avoid exploits as MITM (man-in-the-middle), spoofed attacks and session hijacking.

(ii) Patching and Updating – The concept of timely patching and updating of systems are enforced by most of the standards organizations. But, the proliferation of VMs in the organization adds complexity to the patch control process. This means that we not only patch and update the host OS promptly, but every virtual machines needs the same patching schedule.

(iii) VM Maintaining Backups – Perform image backup frequently for all production VMs. This will aid recovery of both individual files or the complete server image. Protection of the physical backup is also a part of best practices. This includes protection of the data stream of the backup.

(iv) Enabling Perimeter Defense on the VM – The perimeter defense devices are some of the oldest and most established ways of enforcing the security policy, by regulating data traffic ingress and egress.

(v) VM Hardening the Host Operating System – The hardening techniques must be implemented to maintain the security posture of the underlying technology. Some of these techniques include are –

(a) After testing on a non-production unit, patch and update the host regularly.

(b) Individually firewall should be host.

(c) Use strong passwords, like lengthy, hard to guess passwords with letters, numbers, small character combinations and symbol combinations, and change them often.

(vi) VM Implementing File Integrity Checks – File integrity checking is the process of verifying that the files retain the proper consistency, and serves as a check for intrusion into the system.

Q.37. Describe the VM-specific security techniques.

Ans. Some important VM-specific security techniques are as follows –

(i) Root Secure the VM Monitor – The VM monitor should be root secure because many operating systems can be compromised through privilege escalation.

(ii) Implement only One Primary Function per VM – It is very difficult for hacker's ability to compromise multiple system components if the VM is implemented with one primary function per virtual server or device.

(iii) VM Harden the Hypervisor – It is difficult to attention on the hypervisor as an attack vector, and strive to ensure that the hypervisor is deployed securely.

(iv) Firewall any Additional VM Ports – VM may open multiple ports linked to the host's external IP address, besides the usual ports opened by the host. The host system should be independently firewalled with a minimum of access allowed.

(v) VM Harden the Host Domain – The Host Domain of the host Linux OS system must be secure, before securing any virtual machine.

A compromise of the Host Domain makes compromising the Guest Domains a simple task. Thus steps should be taken to reduce the attack surface of the Host Domain. These include the following –

(a) Remove unnecessary groups and accounts.

(b) Remove unnecessary files, binaries, and libraries.

(c) Disable unnecessary services.

(d) Firewall network access to the host.

(e) Install monitoring or Host intrusion detection systems.

Q.38. Discuss secure execution environment and communications in cloud computing.

Ans. **Secure Execution Environment** – One of the difficult tasks in cloud computing is the configuration of computing platforms for secure execution. In many cases, it is not performed well due to several involved parameters. This offers opportunities for malware to exploit vulnerabilities, like downloading code embedded in data and having the code run at a high privilege level.

The main difficulty of creating a secure execution environment in cloud computing is transferred from the client to the cloud provider. However, protected data transfers are done using strong authentication mechanisms. In essence, the client's port to the cloud may offer an attack path in the absence of properly provisioned security measures. Hence, computations and data exchanges are done in a secure environment to assure the client.

In secure execution of code, another great concern is the extreme use of unsafe programming languages like C and C++ in place of more secure languages like object-oriented C# and object-oriented Java.

Secure Communications – Secure cloud communications encompasses the structures, transport formats, transmission methods, and security measures that give confidentiality, integrity, availability, and authentication for transmissions over public and private communications networks. Secure cloud computing communications makes sure the following –

(i) Confidentiality – It makes sure that only those who are authorized to access data can retrieve it.

(ii) Integrity – It makes sure that data has not been altered because of an accident or malice.

(iii) Availability – It makes sure that data is accessible to authorized users when required.

Q.39. What is VPN ? Also discuss its types. [R.G.P.V., May 2018 (MCA)]

Ans. A virtual private network (VPN) is formed by constructing a secure communications link between two nodes by using the properties of a point-to-point private link. A VPN is used to make simple secure remote access into the cloud, establish a secure data tunnel within a network, or securely connect two networks together.

The ***tunnel*** is the portion of the link in which the private data is encapsulated. Data is encapsulated with a header that gives routing information to emulate a point-to-point link. Mostly, the encryption of data is done to achieve confi- dentiality. This encrypted link part is viewed as the actual virtual private network connec- tion. A common VPN configuration with example IP addresses for remote access into a company's intranet through the Internet is shown in fig. 4.3. Address 192.168.123.3 denotes the company's router.

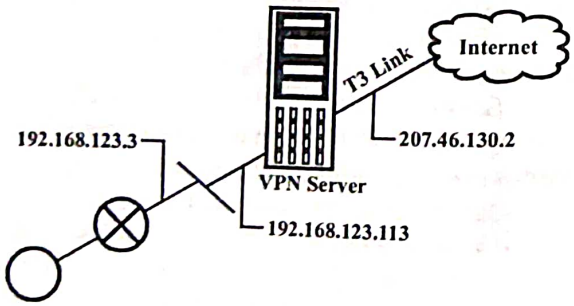


Fig. 4.3 VPN Configuration

There are two general types of VPNs relevant to cloud computing namely, remote access VPNs and network-to-network VPNs. These VPNs types are discussed as follows –

(i) Remote Access VPNs – To maintain confidentiality and integrity, a VPN can be configured to offer remote access to company's resources over the public Internet. This configuration allows the remote user to use local ISP to access the Internet without forcing the user to create a long-distance or 800 call to a third-party access provider. The VPN software establishes a virtual private network between the dial-up user and the company's VPN server across the Internet using the connection to the local ISP. A remote user VPN connection is shown in fig. 4.4.

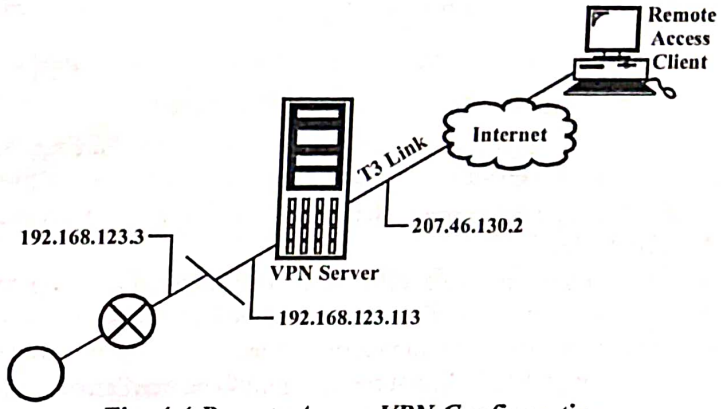


Fig. 4.4 Remote Access VPN Configuration

(ii) Network-to-network VPNs – A common use of a VPN is to connect two networks, perhaps the main company's LAN and a remote branch office LAN, through the Internet. A VPN connection can be either dial up or dedicated lines. The connection to the local ISP is used by the VPN software to establish a VPN tunnel between the branch office router and the company's hub router across the Internet. A remote branch office connected to the company's main office using a VPN tunnel through the Internet is shown in fig. 4.5.

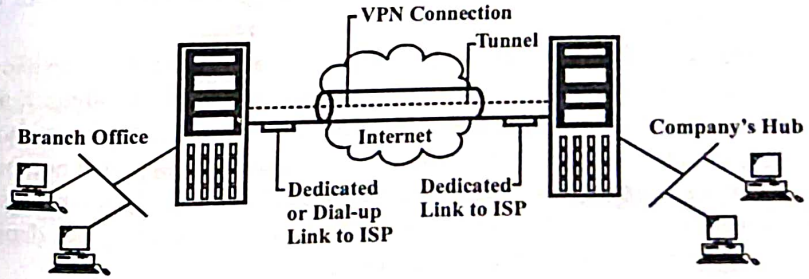


Fig. 4.5 A Network-to-network VPN Configuration

Q.40. Explain about VPN tunneling in detail.

Ans. A method of transferring data from one network to another network by encapsulating the packets in an additional header is known as ***tunneling***.

The extra header gives routing information so that the encapsulated payload can travel in the intermediate networks. This is depicted in fig. 4.6.

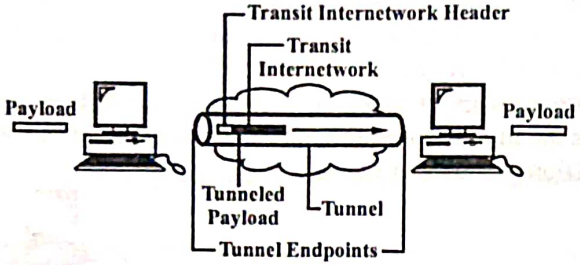


Fig. 4.6 A VPN Tunnel and Payload

Both the tunnel client and the tunnel server must be utilizing the same tunneling protocol to establish a tunnel. Tunneling technology depends on either a Layer 2 or a Layer 3 tunneling protocol. Both layers correspond to the OSI reference model.

Tunneling, and the use of a VPN, is not con- sidered as a substitute for encryption and de- cryption. The strongest possible encryption should be used within the VPN itself, and tunneling should serve only as a convenience in cases where a high level of security is required.

IPSec is a popular tunneling protocol for network-to-network connectivity. IPSec encapsulates IP packets in an extra IP header. It functions at the network layer of OSI model and permits multiple simultaneous tunnels. IPSec can encrypt and authenticate IP data.

Q.41. Write short note on lightweight directory access protocol.

Ans. A more efficient version of the DAP is the lightweight directory access protocol (LDAP). LDAP servers communicate through referrals. It sends a referral to the requesting directory, if it gets a directory with the needed entry.

A standard format is given by LDAP for accessing the certificate directories. These directories offer public keys and corresponding X.509 certificates for the enterprise and are stored on network LDAP servers. A directory has information like individual's names, addresses, phone numbers, and public key certificates. The standards under X.500 specify the protocols and information models for computer directory services that do not depend on the platforms and other related entities.

UNIT 5

ISSUES IN CLOUD COMPUTING, IMPLEMENTING REAL TIME APPLICATION, QoS ISSUES IN CLOUD, DEPENDABILITY, DATA MIGRATION, STREAMING IN CLOUD, CLOUD MIDDLEWARE

Q.1. Write down the issues in cloud computing.

Ans. Issues in cloud computing are as follows –

(i) Privacy – Cloud computing utilizes the virtual computing technology, users' personal data may be scattered in various virtual data centers rather than stay in the same physical location, users may leak hidden information when they care accessed cloud computing services. Attackers can analyze the critical task depend on the computing task submitted by the users.

(ii) Reliability – The cloud servers also experience downtimes and slowdowns as our local server.

(iii) Legal Issues – Worries stick with safety measures and confidentiality of individual all the way through legislative levels.

(iv) Compliance – Numerous regulations pertain to the storage and use of data requires regular reporting and audit trails. In addition to the requirements to which customers are subject, the data centers maintained by cloud providers may also be subject to compliance requirements.

(v) Freedom – Cloud computing does not allow users to physically possess the storage of the data, leaving the data storage and control in the hands of cloud providers.

(vi) Long-term Viability – You should be sure that the data you put into the cloud will never become invalid even your cloud computing provider go broke or get acquired and swallowed up by a larger company.

Q.2. Write a short note on Google AppEngine.
[R.G.P.V., May 2018, May 2019 (MCA), Nov. 2019 (MCA)]

Ans. Google AppEngine is a Platform-as-a-Service implementation. It offers services for developing and hosting expandable Web applications. It is a distributed and scalable runtime environment that uses Google's distributed infrastructure to scale out applications handling several requests by allocating more computing resources to them and balancing the load among them. The completion of runtime is done by a collection of services permitting developers to design and implement applications that scale on AppEngine. The languages like Java, Python, and Go are used by developers, to develop applications. AppEngine constantly meters application's usage of Google resources and services. It bills users when their applications trespass free quotas.

Q.3. Describe the major cloud features of Google application engine. ***[R.G.P.V., Dec. 2013 (BE), Dec. 2016 (BE)]***

Ans. Google application engine supports the following major features –

(i) Persistent storage, with query access sorting and transaction management features.

(ii) Scheduled tasks for triggering events at specified times or regular intervals.

(iii) Asynchronous task queues for performing work outside the scope of a request.

(iv) Automatic scaling and load balancing.

(v) One of either two runtime environments – Java or Python.

(vi) Authentication using Google Accounts API.

(vii) Dynamic Web services based on common standards.

(viii) A client side development environment for simulating Google application engine on your local system.

(ix) Integration with other Google cloud services and APIs.

Q.4. Discuss the runtime environment component of Google AppEngine.

Ans. The execution context of applications hosted on AppEngine is represented by the runtime enviornment. The runtime comes into existence when the request handler begins to execute and terminates once the handler has completed.

(i) Sandboxing – A key responsibility of the runtime environment is to offer application environment with an isolated and protected context where they can run without forming a threat to the server and without being affected by other applications. That is, applications are offered by it with a sandbox.

(ii) Supported Runtimes – At present, developers can develop AppEngine applications in Java, Python, and Go.

At present, AppEngine supports Java 6. Developers utilize the common tools for Web applications development in Java, like the JSP and the

applications interact with the enviornment by means of Java Servlet standard. Also, Java libraries provides access to AppEngine services Java libraries reveal specific interfaces of provider specific implementations of given abstraction layer.

An optimized interpreter supporting python 2.5.2 offers support for Python. The runtime environment supports the Python standard library but some modules implementing harmful operations have been eliminated and try to import such modules or call specific methods produce exceptions. AppEngine contains a collection of libraries connecting applications to AppEngine services with respect to application development.

The applications developed with the Go programming language are hosted and executed in AppEngine. r58.1 is the recent version of Go. Some of the modules have been deleted or produce a run-time exception.

Q.5. Discuss the compute services offered by AppEngine.

Ans. AppEngine has services that facilitate the execution of computations that are off-bandwidth, i.e., that cannot be contained within the time frame of the Web request handling. These are –

(i) Task Queues – Applications can submit a task for later execution using task queues. Task queues are useful for long computations that cannot finish within the maximum response time of a request handler. Users can have maximum 10 queues to execute tasks at a configurable rate.

A Web request to a given URL defines a task. The request handler is invoked by the queue by passing the payload as part of the Web request to the handler. The request handler performs the task execution. In case of failure, the queue re-executes the task in order to avoid that transient failures stop the task from a successful completion.

(ii) Cron Jobs – It might happen that the needed operation requires to be performed at a particular time of the day, which does not coincide with the time of the Web request. In this situation, cron jobs service may be used to schedule the needed operation at the specified time. The service works same as task queues, but invokes the request handler specified in the task at a given time. In case of failure, the service does not re-execute the task.

Q.6. Explain a user view of Google AppEngine with suitable block schematic. ***[R.G.P.V., June 2017 (MCA)]***

Ans. Google AppEngine platform architecture are divided into four components –

(i) Infrastructure – Web applications can be hosted by AppEngine and its primary function is to serve users requests efficiently.

(ii) Runtime Environment – Refer to Q.4.

(iii) Storage – There are three different level of storage – in memory cache, storage for semi-structured data and long-term storage for static data.

(a) Static File Servers – Web applications comprised of static and dynamic data. Static data is mostly constituted through the elements which express the graphical application layout (CSS files, sound files, java script files and plain html files) and data files. Dynamic data is a result of the application logic and the interaction with the user.

(b) Data Store – A service permitting developers to store semi-structured data is referred to the data store.

(iv) Application Services – Application hosted on AppEngine consider the most from the services made available by the runtime environment. These services simplify most of the common operations which are done in web applications.

(v) Compute Services – Refer to Q.5.

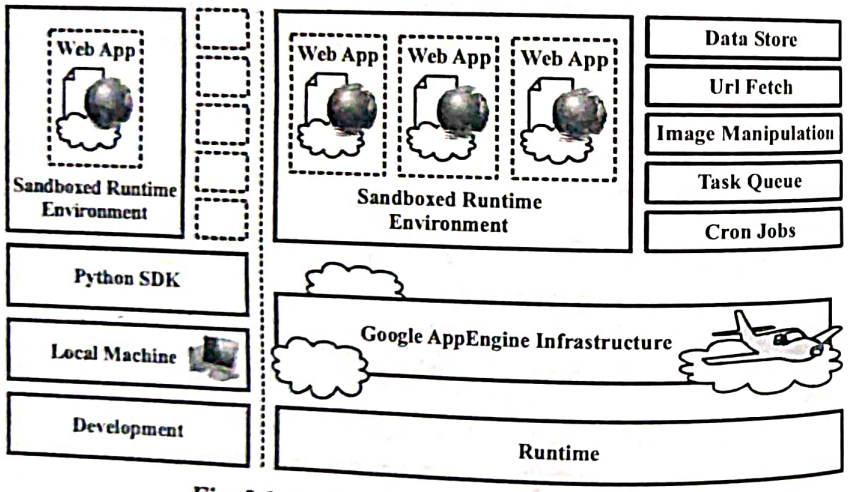


Fig. 5.1 Google AppEngine Architecture

Q.7. What are the programming supports of Google AppEngine ? Also explain Google File System. *[R.G.P.V., June 2015 (MCA)]*

Ans. Google AppEngine currently supports applications written in Java and in Python, although there are plans to extend support to more languages in the future. The service is meant to be language-agnostic. A number of Java virtual machine languages are complicant with GAE, as are several Python Web frameworks that support the Web Server Gateway Interface (WSGI) and CGI. Google has its own Webapp framework designed for use with GAE. The AppScale open source framework also may be used for running applications on GAE. To encourage developers to write applications using GAE, Google

allows for free application development and deployment upto a certain level of resource consumption.

Also refer to Q.4 (Unit-III).

Q.8. What do you mean by quality of service (QoS) ?

Ans. Users of Internet network is increasing day-by-day, network requirement also increases to achieve good performance. Therefore, many online services need a very large bandwidth and network performance. Network performance is the element that disquiet the users and service providers. Internet service providers should bring new technologies to provide the best services before competitors strike them.

Quality of Service refers to the ability of networks to attain maximum bandwidth and handle other network elements like latency, error rate and uptime. Quality of Service include the management of other networks resource by allocating priorities to specific type of data (audio, video and file).

Basic implementation of QoS need three major component such as –

(i) QoS within one network element.

(ii) QoS policy and management functions to control end-to-end traffic across network.

(iii) Identification techniques for coordinating QoS from end-to-end between network elements.

Q.9. Describe the issues and challenges of QoS in cloud computing.

Ans. The aim of cloud computing is to effectively exploit the shared pool of various resources so as to optimize the computing. To maintain the cloud platforms, cloud resources and services various issues and challenges are associated.

Issues – There a major issue associates with management of cloud services results to the catastrophe. With the increasing trend of the cloud services it become more difficult to investigate the QoS for cloud. The prime concern is the security and privacy during the transmissions of the resources in cloud. The study shows that many companies like facebook, Amazon, Google are bit concerned about the data which should be kept confidential to share with other companies with high bandwidth rate with having less delay. Choice of database is a crucial component in the software stack of many cloud hosted applications. The various existential issues associated with the flood server are –

(i) Managing and ensuring application in QoS
(ii) Cost
(iii) Increasing services for users
(iv) Slow applications when hosted on sever with more errors

(v) Guaranteed own SLA's
(vi) No data limits
(vii) Performance of the applications
(viii) System backlog.

The cloud applications need to be managed properly online so that it can serve best to its clients. This may lead to various viz delay, jittter or packet loss associated with real-time applications. Traditionally, fault tolerance software's was quiet costlier but got reduced when cloud services came into the existence.

Challenges – The main challenge is to solve scalability and the dependability issues in managing the QoS in cloud computing. fig. 5.2 shows the CloudDB framework for a management of the application-defined service level agreements (SLA) for cloud hosted database. The below mentioned architecture manages the input and output of the database in cloud applications. The SLA checker checks the results of monitoring module and make comparisons against the application defined SLA and reports if SLA is violated. Basically, it checks how many SLA's has been violated.

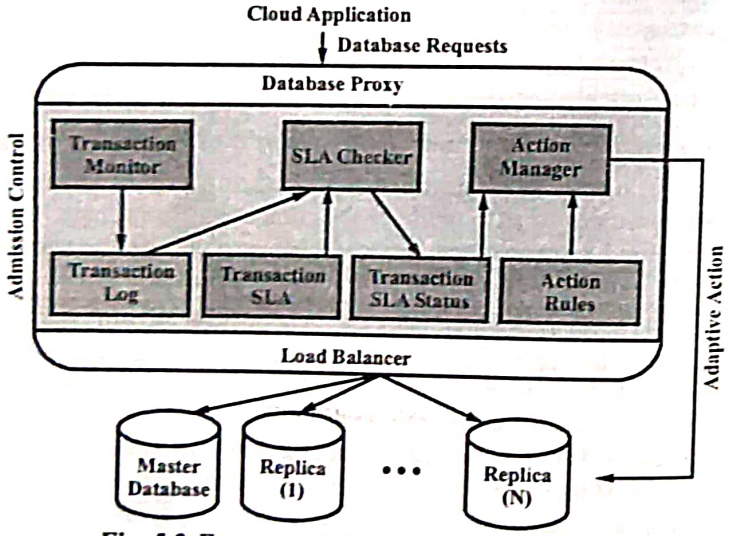


Fig. 5.2 Framework for CloudDB AutoAdmin

Q.10. Brief note on dependability techniques in cloud computing.

Ans. In general, the faults can be classified in different ways depending on the nature of the system. Since, we are interested in typical faults that appear as failures to the end users, we classify the faults into two types similarly to other distributed systems. First, crash faults that cause the system components to completely stop functioning or remain inactive during failures

(e.g., power outage, hard disk crash), and second, byzantine faults that leads the system components to behave arbitrarily or maliciously during failure, causing the system to behave unpredictably incorrect.

The most widely adopted methods to achieve fault tolerance against crash faults and byzantine faults are as follows –

(i) Checking and Monitoring – The system is constantly monitored at runtime to validate, verify and ensure that correct system specifications are being met. This technique, while simple, plays a key role in failure detection and subsequent reconfiguration.

(ii) Checkpoint and Restart – The system state is captured and saved based on pre-defined parameters (e.g., after every 1024 instructions or every 60 seconds). When the system undergoes a failure, it is restored to the previously known correct state using the latest checkpoint information (instead of restarting the system from start).

(iii) Replication – Critical system components are duplicated using additional hardware, software and network resources in such a way that a copy of the critical components is available even after a failure happens. Replication mechanisms are mainly used in two formats – active and passive. In active replication, all the replicas are simultaneously invoked and each replica processes the same request at the same time. This implies that all the replicas have the same system state at any given point of time (unless designed to function in an asynchronous manner) and it can continue to deliver its service even in case of a single replica failure. This method is also called as hot standby. In passive replication, only one processing unit (the primary replica) processes the requests while the backup replicas only save the system state during normal execution periods. Backup replicas take over the execution process only when the primary replica fails. This method is called as cold standby. The N + M technique of adding M standby hosts to spares for N working hosts to accommodate up to M failures is the most popular solution.

Q.11. Write short note on data migration.

Ans. Data migration to a cloud computing environment is in many ways an exercise in risk management. Both qualitative and quantitative factors apply in an analysis. The risks must be carefully balanced against the available safeguards and expected benefits, with the understanding that accountability for security remains with the organization. Too many controls can be inefficient and ineffective, if the benefits outweigh the costs and associated risks. An appropriate balance between the strength of controls and the relative risk associated with particular programs and operations must be ensured. Moreover, if data migration is not done systematically and properly, it can give rise to problems concerning data and cloud security of company's assets that primarily

comprise of data. Thus, hiring cloud providers having sound experience about the field with ample knowledge and skill sets becomes vital for managing cloud more effectively and efficiently.

For example – Suppose an XYZ company wants to shift its data to cloud storage for increased uptime and scalability, it goes to cloud service provider for performing such functions. Now, the cloud provider starts initializing steps for data transfer to cloud, but in between face problems like data crash or unauthorized access by third parties. This is where the problem lies. The proprietor of data that hired cloud manager would not only face reputation losses but also monetary losses. Similar case was experienced when Amazon cloud failure happened and several business suffered immense losses due to it. Thus, securing data remains an utmost priority of cloud managers to prevent global cloud security threats that also include cross-border security concerns.

Some characteristic of data migration are as follows –

(i) Commercial relation exists between clouds.

(ii) Transmission of mass data.

(iii) Many workers which execute transmission process concurrently.

Q.12. How can we improve video quality in cloud streaming ?

Ans. Some approaches to improve video quality of smart phones are as follows –

(i) P2P Live Video Streaming – Cloud-based P2P Live Video Streaming Platform (Cloud PP) that uses public cloud servers to construct an efficient and scalable video delivery platform with Scalable Video Coding (SVC) technology. The cloud server behaves like a SVC extractor, enabling a very large number of clients to receive live video streams at the same time by dynamically arranging available resources based on the streaming quality requested by clients.

(ii) Asymmetric Graphics Rendering – This method significantly reduces the video encoding bit rate needed for a certain video quality, thereby making it easier to transmit the video over wireless network. It is possible to set appropriate graphics rendering parameters according to network constraints, such that the user experience can be maintained to a high level.

(iii) Scalable Video Coding – The cloud server behaves like a SVC extractor, enabling a very large number of clients to receive live video streams at the same time by dynamically arranging available resources based on the streaming quality requested by clients. SVC standardizes the encoding of a high-quality video bit stream that also contains one or more subset bit streams. The subset bit stream can represent a lower spatial resolution (smaller screen), lower temporal resolution (lower frame rate), or lower quality video signal.

(iv) Adaptive Mobile Video Streaming (AMoS) – In cloud we use user-Adaptive Mobile Video Streaming (AMoS) and the User Behaviour Oriented Video Pre-Fetching (UBoP). This method reduces the traffic using SVC for adjust streaming. For distributing video in proper way in mobile network used private agent. It shows the social interaction between the mobile users. Video quality based on feedback of link quality. Result shows that the cloud can effectively provide the video streaming and video sharing on network.

(v) Rendering Adaptation Technique – In this technique graphic rendering work on cloud instead of mobile devices. This is based on bit rate and computation load.

Q.13. What do you mean by middleware ?

Ans. Middleware is an important class of technology that is serving to decrease the cycle-time, level of effort, and complexity associated with developing high-quality, flexible, and interoperable distributed systems. When implemented properly, middleware can help to shield developers of distributed systems from low-level, tedious, and error-prone platform details, such as socket-level network programming. It was invented in an attempt to help simplify the software development of distributed computing systems, and bring those capabilities within the reach of many more developers than the few experts at the time who could master the complexities of these environments. Complex system integration requirements were not being met from the application perspective, where it was too hard and not reusable, or the network or host operating system perspectives, which were necessarily concerned with providing the communication and end system resource management layers, respectively. One also finds business intelligence, content and collaboration tools, as well as portal capabilities that allow connections to customers and partners enabled at the middleware level, middleware and middleware-based architectures. Middleware is systems software that resides between the application and the underlying operating systems, network protocol stacks, and hardware.

Its primary role is to (i) functionally bridge the gap between application programs and the lower-level hardware and software infrastructure in order to coordinate how parts of applications are connected and how they interoperate and (ii) enable and simplify the integration of components developed by multiple technology suppliers.

Q.14. Write short note on cloud middleware.

Ans. Datacenters running a cloud environment often enclose a large number of machines that are connected by a high-speed network. Users access sites hosted by the cloud environment through the public Internet. A site is typically accessed through a URL that is translated to a network address through a global directory service, such as DNS. A request to a site is routed through the

Internet to a machine inside the datacenter that either processes the request or forwards it.

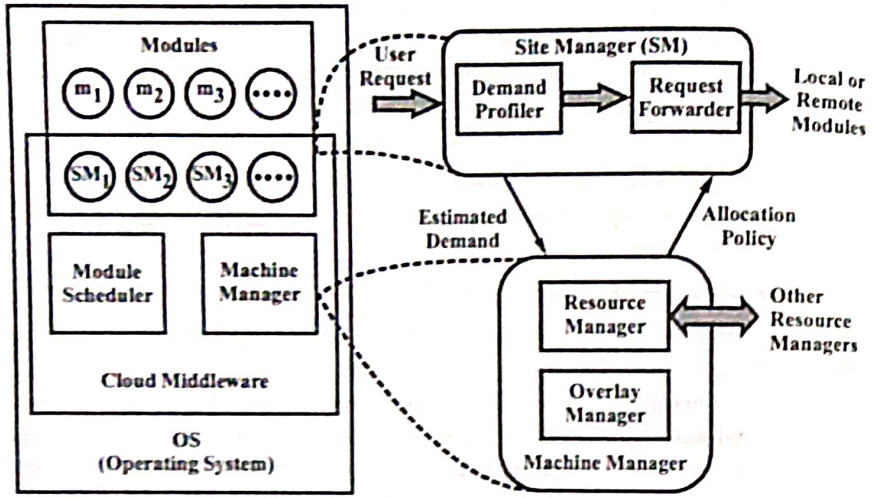


Fig. 5.3 Cloud Middleware

Q.15. Give examples of middleware.

Ans. Middleware is sometimes used in a similar sense to a software driver, an abstraction layer that hides detail about hardware devices or other software from an application. The Android environment uses the Linux operating system at its core, and also provides an application framework that developers incorporate into their applications. The Android middleware layer also contains the Dalvik virtual machine and its core Java application libraries. Game engine software such as Gamebryo and Renderware are sometimes described as middleware, because they provide many services to simplify game development. In simulation technology, middleware is generally used in the context of the high level architecture (HLA) that applies to many distributed simulations. The QNX operating system offers middleware for providing multimedia services for use in automobiles, aircraft and other environments. Multimedia Home Platform (DVB-MHP) is an open middleware system standard designed by the DVB project for interactive digital television.

MOBILE CLOUD COMPUTING, INTER CLOUD ISSUES, A GRID OF CLOUDS, SKY COMPUTING, LOAD BALANCING, RESOURCE OPTIMIZATION, RESOURCE DYNAMIC RECONFIGURATION

Q.16. What is mobile cloud computing ?

Ans. Several definitions of mobile cloud computing (MCC) are available.

For example, mobile cloud computing is defined as "a rich mobile computing technology that leverages unified elastic resources of varied clouds and network technologies toward unrestricted functionality, storage, and mobility. It serves a multitude of mobile devices anywhere anytime through the channel of Ethernet or Internet regardless of heterogeneous environments and platforms based on the pay-as-you-use principle".

Another definition of mobile cloud computing is that – "Mobile cloud computing at its simplest, refers to an infrastructure where both the data storage and data processing happen outside of the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and MC to not just smartphone users but a much broader range of mobile subscribers".

Q.17. What are the advantages of mobile cloud computing ?

Ans. The advantages of mobile cloud computing are as follows –

(i) Compute and Storage Efficiency – By off-loading demanding workloads and large data to the cloud, the mobile device can limit the amount of processing power and data storage that it requires.

(ii) More Powerful Mobile Applications – Since the mobile device now has access to a powerful cloud on the back end, we have the potential to create more powerful mobile applications than previously possible.

(iii) Energy Efficiency – Much of the resource-intensive work in mobile applications can be off-loaded to the cloud, which means that mobile clients can focus more on reducing energy consumption without trading off on performance.

(iv) Thin Mobile Clients – Less resource demands on the mobile client means that we can build less powerful mobile devices that achieve better overall performance when coupled with a cloud platform. This gives us the ability to "dumb down" the mobile clients, to the extent that they only handle user interaction and off-load all application work and data to the cloud.

Q.18. Describe the features of mobile cloud computing.

Ans. The primary features of mobile cloud computing as shown in fig. 5.4 are as follows –

(i) Auto Resource Provision and De-provision – Mobile clouds enable auto resource provisions and de-provisions of cloud computing resources, network resources, and mobile device resources.

(ii) Scalability – In mobile clouds, scalability includes three dimensions –

(a) Cloud scalability
(b) Network scalability
(c) Mobile scalability in terms of mobile users and devices.

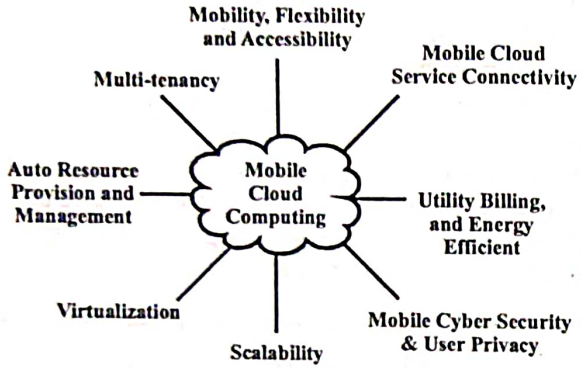


Fig. 5.4 Primary Features of Mobile Clouds

(iii) Mobility, Flexibility, and Accessibility – Mobile clouds enable mobile users to access mobile cloud applications and services anytime and anywhere with personal accessibility.

(iv) Mobile Cloud Service Connectivity – Mobile clouds offer well-defined connectivity APIs and protocols to enable easy and secured connectivity between different networks, and standards, and third-party software and systems.

(v) Virtualization – Three types of virtualizations can be supported in mobile clouds –

(a) Network virtualization
(b) Cloud virtualization for various computing resources
(c) Mobile devices and resources.

(vi) Multi-tenancy – This feature allows single mobile cloud software instance to serve multiple mobile tenants on a wireless Internet or heterogeneous networks.

(vii) Mobile Cyber Security and Privacy – This refers to the body of security capabilities, technologies, processes and practices designed to protect mobile devices, heterogeneous networks (both wireless network and Internet), cloud servers, mobile application service programs, and data from attack, damage or unauthorized access.

(viii) Mobile Utility Billing and Energy Efficient – This refers to the provided mobile-based utility models, i.e., meter-based, volume based, and subscription-based for service billing.

Q.19. What is intercloud ? Explain architecture of intercloud.
[R.G.P.V., Dec. 2013 (MCA)]

Ans. The term intercloud is used interchangeably to express the concept of cloud federation. Intercloud expresses a composition of clouds that are interconnected using open standards to offer a universal environment for using cloud computing services. Intercloud represents a cloud of clouds and hence expresses the same concept of federating together clouds pertaining to different administrative organizations.

Intercloud Architecture – The intercloud architecture consists of two elements – cloud exchange and cloud coordinator. This is shown in fig. 5.5.

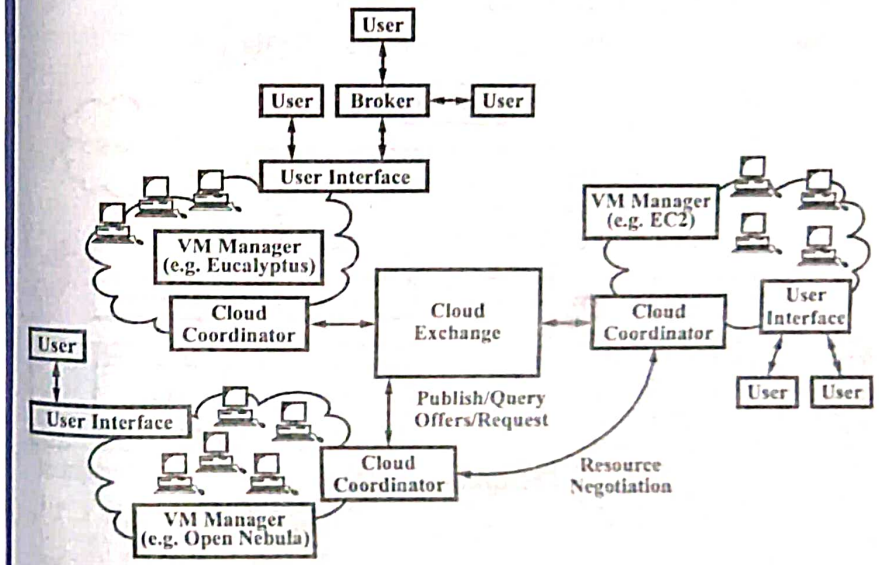


Fig. 5.5 Intercloud Architecture

(i) CloudExchange – It is the market-making component of the architecture. It provides services that permit providers to detect each other in order to directly trade cloud assets, as well as permits parties to registry and execute auctions. In the first case, CloudExchange works as a directory service for the federation. In the second case, CloudExchange executes the auction. CloudExchange implements a web service based interface that permits data centers to join and leave the federation for providing such services to the federation.

(ii) CloudCoordinator – It manages domain-specific issues pertaining to the federation. This is available on each party that wishes to join the federation. It contains front-end components and back-end components. The interaction of front-end components takes place with the CloudExchange

and with other coordinators. The former permits data centers to mention their offers and needs, while the latter permits the coordinator to learn about the current state of the data center to determine whether actions from the federation are needed or not. Hence, when the coordinator finds that additional resources are needed by the data center, it initiates the discovery process of potential providers. As soon as the potential providers are found and the interested one is chosen, the coordinator meets the remote coordinator and communicates. Likewise, when CloudCoordinator finds that local resources are in use, they can mention an offer for resources in the CloudExchange, or they can search for matches among needs registered in the exchange service.

Q.20. Describe the classification of intercloud.

Ans. The classification of intercloud is shown in fig. 5.6.

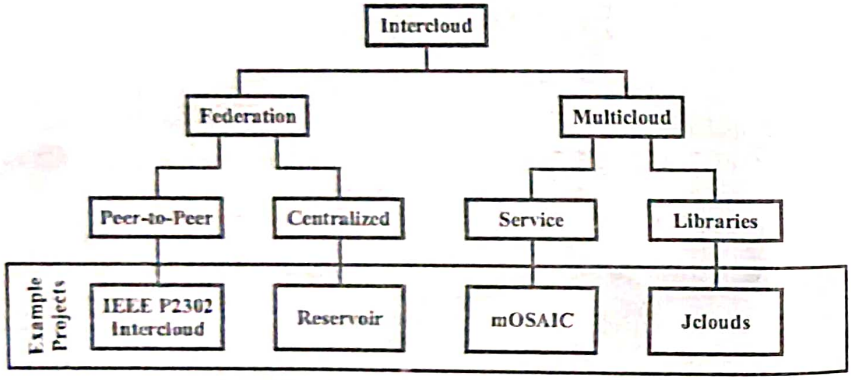


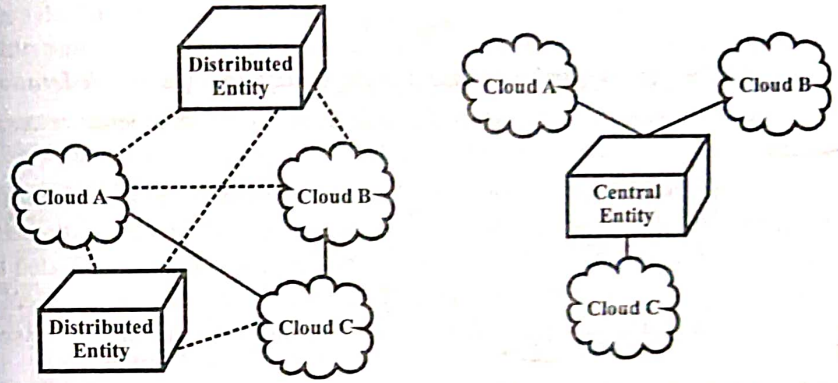
Fig. 5.6 Classification of Interclouds

(i) Federation Cloud – A federation cloud is an intercloud where a set of cloud providers willingly interconnect their cloud infrastructures in order to share resources among each other. The cloud providers in the federation voluntarily collaborate to exchange resources. This type of intercloud is suitable for collaboration of governmental clouds (clouds owned and utilized by nonprofit institution or government) or private cloud portfolios (cloud is a part of a portfolio of clouds where the clouds belong to the same organization). Federation clouds takes care of consistency and access controls if more than one free geographically distinct clouds share either authentication, files, computing resources, command and control or access to storage resources. Types of federation clouds are peer-to-peer and centralized clouds.

(a) Peer-to-peer Intercloud Federation – Clouds collaborate directly with each other but may use distributed entities for directories or brokering. Clouds communicate with each other and negotiate directly without mediators. Peer-to-peer intercloud federation is depicted in fig. 5.7 (a). The

intercloud projects that use peer-to-peer federation are RESERVOIR (resoruces and services virtualization without barriers project), open cirrus, OPTIMIS, Arjuna agility and global intercloud by Bernstein et al.

(b) Centralized Intercloud Federation – Clouds use a central entity to perform or facilitate resource sharing. The central entity acts as a storehouse where the available cloud resources are registered. Centralized intercloud federation is depicted in fig. 5.7 (b). The intercloud projects that use centralized intercloud federation are intercloud, contrail, dynamic cloud collaboration (DCC) and federated cloud management.



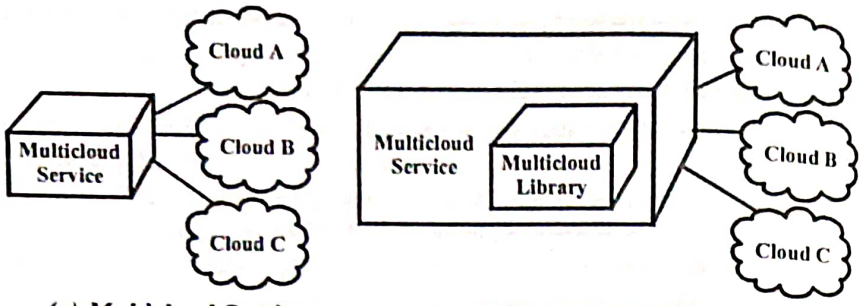
(a) Peer-to-peer intercloud Federation *(b) Centralized Intercloud Federation*

Fig. 5.7

(ii) Multicloud – In a multicloud, a client or service uses multiple independent clouds. A multicloud environment has no volunteer interconnection and sharing of the cloud service providers' infrastructures. Managing resource provisioning and scheduling is the responsibility of client or their representatives. This approach is used to utilize resources from both governmental clouds and private cloud portfolios. Types of multicloud are services and libraries.

(a) Multicloud Service – Clients access multiple clouds through a service. A service is hosted by the cloud client either externally or in-house. The services contain broker components. The multicloud service is depicted in fig. 5.8 (a). The intercloud projects that use multicloud services are OPTIMIS, contrail, mOSAIC, STRATOS and commercial cloud management systems.

(b) Multicloud Libraries – Clients develop their own brokers by using a unified cloud API as a library. interclouds that use libraries facilitate the usage of clouds in a uniform way. Multicloud libraries is depicted in fig. 5.8 (b). Examples of several multicloud libraries are Java library JClouds, Python library Apache LibClouds, Ruby library, Apache DeltaCloud, PHP library SimpleCloud, ApacheNuvem.



(a) Multicloud Service *(b) Multicloud Library*

Fig. 5.8

Q.21. Discuss the various aspects in the realization of cloud federation.

Ans. The aspects involved in the realization of cloud federation are as follows –

(i) Standards – The role of standards is to build a platform for interoperation that goes beyond adhoc aggregations and private settlements between providers. The realization of an open organization is facilitated by standardized interfaces and protocols. The benefits are mainly technical – standards make easy the development of software and services that interconnect systems.

(ii) Security – By using large computing infrastructures that eventually host applications, services, and data, cloud computing helps the development of expandable systems. Here, security arrangements form a fundamental need that cannot be neglected. Security management is even more difficult in the case of cloud federation, where confidential information is dynamically moved across different cloud computing vendors.

In a cloud scenario, key elements identified in the management of security are – availability management, access control, patch management, vulnerability management, configuration management, incident response, system use and access monitoring.

(iii) Legal Issues – Apart from the technical difficulties needed in making cloud computing occur, legal issues pertaining to access rights, privacy, and control are peculiar to cloud computing.

Q.22. What do you mean by the term elasticity related to federation cloud ? ***[R.G.P.V., June 2017 (BE)]***

Ans. Elasticity refers to the capability of system to adapt to workload depend on on-demand provisioning and de-provisioning of resources. Hence at every time, current resources try to fulfil the current demand. Elasticity and scalability are different from each other.

Q.23. What kind of standards and protocols can be used to achieve interoperability in a cloud federation ? ***[R.G.P.V., Dec. 2014 (BE)]***

Ans. Following are some standards and protocols used to achieve interoperability in a cloud federation –

(i) Open Cloud Manifesto – This is the first step towards the realization of a cloud interoperability platform. The manifesto has been drafted in 2009 as a result of the coordinated activity of different cloud vendors and at the state writing lists more than 400 cloud computing services providers supporting the vision it embodies. More than proposing standards, the manifesto is a declaration of intent, endorsed by commercial players in the field of cloud computing, in realizing an interoperable and open cloud computing platform.

(ii) Distributed Management Task Force (DMTF) – It is an organization with more than 4000 active members, 44 countries, and nearly 200 organizations. DMTF is the industry organization leading the development, adoption and promotion of interoperable management standards and initiatives. With specific reference to cloud computing, the DMTF has introduced the open virtualization format and supported several intitiatives for interoperable cloud technologies such as the open cloud standards incubator, the cloud management working group and the cloud audit data federation working group.

(iii) Open Cloud Computing Interface (OCCI) – It is an open organization constitutes a set of specifications driven by the community and delivered through the open grid forum. These specifications define protocols and API for various types of management tasks. Initially conceived to create a remote management API for IaaS type services, it has evolved into a wider set of APIs focusing on integration, portability and interoperability.

(iv) Cloud Data Management Interface (CDMI) – It is a specification of a functional interface that applications will use to create, retrieve, update and delete data elements from the cloud. This interface also provides facilities for discovering the properties of a given cloud storage offering CDMI has been proposed by the cloud storage technical working group of the storage network industry association, which is an association promoting standards in the management of IT information with a particular focus on data storage. Storage network industry association has also produced a reference implementation of the CDMI, thus facilitating the process of quickly producing a working standard by means of the feedback from the community.

Q.24. Discuss the interoperability issues in federated cloud environment. ***[R.G.P.V., June 2017 (BE)]***

Ans. Cloud federation has several interoperability issues –

(i) Manageability – Although most cloud solutions control elasticity yet smart algorithms are required for efficient resource utilization.

(ii) Data Management – Meta-data information is required to support data in the cloud, and update standards are required to guarantee long-term storing and interoperable sharing among multiple service providers.

(iii) Privacy and Security – Legislative issues during data distribution should be addressed and security problems are arise during resource sharing among multiple system must be eliminate.

(iv) Federation and Interoperability – Proprietary method should be replaced by standard data structure and data representation method. To eliminate vendor lock-in, new methods are to be find to help interoperability.

(v) Virtualization and Adaptability – Resource scheduling algorithms are needed to be improve to help cross platform executions. Migrations taking into account sudden increase in demands and rapidly changing workloads.

(vi) Programming Models – New techniques are required to improve application development and deployment. Control on data distribution should be improved.

(vii) Energy Efficiency – Scheduling policies are required to provide green and more efficient resource utilization with less power consumption.

Q.25. What are issues faced in intercloud ?

Ans. The requirements of cloud users often require different resources and the requirements are flexible and unpredictable most of the times. This aspect poses complex problems in provisioning of resources and delivery of application services. The following are the challenges faced in federation of cloud infrastructures –

(i) Application Service Behaviour Prediction – It is important that the system should be able to foresee the demands and the behaviour of the services. Only when it can predict, if can take decisions intelligently to dynamically scale up and down. Prediction and forecasting models must be built. The challenge is to build such models that accurately learn and fit statistical functions suitable to different behaviours. It is more challenging to correlate between different behaviours of a service.

(ii) Flexible Mapping of Services to Resources – It is important to maximize the efficiency, cost-effectiveness and utilization because of high operating costs and energy requirements. The system has to compute the best software and hardware configurations which result in a complex process of mapping services to cloud resources. Mapping of services must guarantee that QoS targets are satisfied along with maximum system efficiency and utilization.

(iii) Economic Models Driven Optimization Techniques – Combinatorial optimization problem is a market driven decision making strategy

which searches the optimal combinations of services and deployment plans. Optimization models must be developed which optimize both resource-centric and user-centric QoS targets.

(iv) Integration and Interoperability – SMEs have a large amount of IT assets like business applications in their premises and may not be migrated to the cloud. Sensitive data in an enterprise also may not be migrated to the cloud for security reasons and privacy. A need related to integration and interoperability arises between assets on premises and the cloud services. Issues related to identity management, data management, and business process orchestration need to be resolved.

(v) Scalable Monitoring of System Components – The components in a federated system are distributed but the techniques employed for system monitoring and management use centralized approaches. Due to concerns of scalability, performance and reliability arising from the management of multiple service queues and large volume of service requests, centralized approaches are not suitable and architectures using service monitoring and management services based on decentralized messaging and indexing models are needed.

Q.26. Write short note on grid computing.

Ans. As an evolution of cluster computing, grid computing was introduced in the early 90s. Grid computing introduced a new approach to access extreme computational power, large storage facilities, and a range of services. Users can use resources similar to other utilities like power, gas, and water. By means of Internet connection, grids initially developed as aggregations of geographically dispersed clusters. A computing grid was a dynamic aggregation of heterogeneous computing nodes.

Q.27. What are the characteristics of grid computing ?

Ans. There are four major characteristics of grid computing as follows –

(i) Each grid node has its own ownership and management according to the distributed nature of the infrastructure. This suggests that no centralized authority is needed to control all the nodes.

(ii) Some open standards are used for the nodes to interact and exchange information.

(iii) Since the users are provided with the possibility to access distributed resources as if they were local, they can easily interact and cooperate together (e.g. exchanging data). Also, the plug and play concept assists on the fly service creation by aggregating codes, data and software components. A good example is the integration of web applications while utilizing multiple CPUs.

(iv) Applications can usually be grid enabled but not all of them will maintain scalability. Moreover, no tools are available to convert applications to fully benefit from the capabilities of a grid.

Q.28. Discuss the advantages of grid computing.

Ans. The advantages of grid computing are as follows –

(i) The grid software acts as the brain behind the whole grid. Thus, it can centrally manage all the policies.

(ii) A high level of scalability is achieved according to the modular plug and play nature of the grid. Any desktop or server can be attached to or removed from the network based on the preference.

(iii) Upgrading does not force a downtime to the network. There are many resources to handle the ongoing tasks or projects while some resources go offline for any purpose.

(iv) The job execution performance increases, especially for the jobs that can be well split into small chunks.

(v) Applications can be split up among servers to be run and then the results can be smoothly combined and analyzed after the whole task is completed. Therefore, large SMP servers may be useless.

Q.29. Discuss some disadvantages of grid computing.

Ans. Some disadvantages of grid computing are as follows –

(i) Many applications should be inevitably upgraded in order to utilize the advantages of the model.

(ii) Since various administrative domains are involved, sharing resources may cause political challenges. Many parties are unwilling to share resources that benefit others.

(iii) Memory-hungry tasks and applications are usually required to be run on a large SMP.

(iv) The interconnection between the nodes (resources) is often required to be fast (gigabit Ethernet).

Q.30. Give the comparison between grid and cloud computing.

Ans. The comparison between grid and cloud computing are as follows –

(i) Construction of the grid is to complete a specified task, such as biology grid, geography grid, national educational grid, while cloud computing is designed to meet general application and there are not grid for a special field.

(ii) Grid emphasizes the "resource sharing" to form a virtual organization. Cloud is often owned by a single physical organization (except the community cloud, in this case, it is owned by the community), who allocates resources to different running instances.

(iii) Grid strives to achieve maximum computing. Cloud is after on-demand computing scale up and down, in and out at the same time optimizing the overall computing capacity.

(iv) Grid aims to provide the maximum computing capacity for a huge task through resource sharing. Cloud aims to suffice as many small-to-medium tasks as possible based on user' real-time requirements. Therefore, multi-tenancy is a very important concepts for cloud computing.

(v) Grid trades re-usability for (scientific) high performance computing. Cloud computing is directly pulled by immediate user needs driven by various business requirements.

As per technical aspects, there are some other comparison between grid and cloud computing –

(i) Usage – Grid computing is very much in use in the academic world and cloud computing is used much more in the corporate sector.

(ii) Platform – Grid consists of smaller grids whereas cloud consists of one massive parallel computing system.

(iii) Allocating Process to Resources – In grid computing, allocating processes to resources is done manually whereas in cloud computing, allocating processes to resources is done automatically.

(iv) Scalability in Application Execution – In grid computing scalability in application execution is sequentially (i.e. user has to wait in a queue) whereas in cloud computing, scalability in application execution is parallel (i.e. user does not have to wait for a node to be freed). So cloud computing is more scalable then grid computing.

Q.31. What is sky computing ? Write down its benefits and challenges.

Ans. Sky computing is an emerging computing model where resources from multiple cloud providers are leveraged to create large scale distributed infrastructures.

Sky computing arises as a metaphor to illustrate a layer above cloud computing, because such dynamically provisioned distributed domains are built over several clouds. It can be described as a management up layer of an environment of clouds, offering variable computing capacity and storage resources with dynamic support to real-time demands. Laying a virtual site over distributed resources, combining the ability to trust remote sites with a trusted networking environment, originates a highly elastic response to incoming requests with a seemingly infinite pool of accessible resources.

Fig. 5.9 shows experiment on the sky in which user-level networks needed for intercloud communication and ViNe enables applications across clouds.

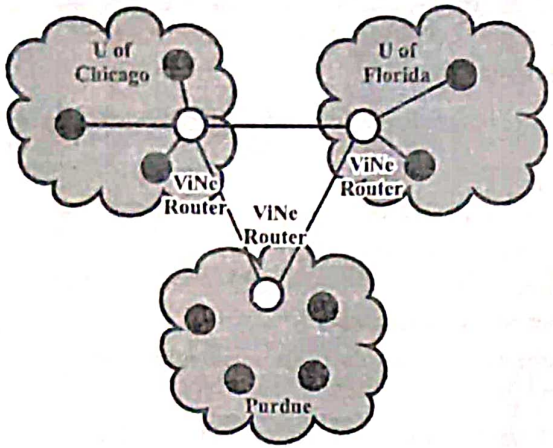


Fig. 5.9 Sky Computing

Benefits – Following are the benefits of sky computing –

(i) Single Networking Context – All-to-all connectivity.

(ii) Single Security Context – Trust between all entities.

(iii) Equivalent to Local Cluster – Compatible with legacy code.

Challenges – Following are the challenges of sky computing –

(i) Intercloud resource creation and management.

(ii) Efficient intercloud communication.

(iii) Efficient distribution of tasks.

(iv) Fault tolerance.

(v) Adaptability to resource dynamicity.

Q.32. Explain in detail about the sky computing architecture.

Ans. The main idea is to create a turn-around model to enable intensive computing in cloud networks. This is hoped to be achieved by enlarging the set of available resources in a way they overcome the problems referred before, like elevated latency between nodes. Also, it must be cross cloud provider in order to combine resources. To achieve this, there must be a structure capable of receiving instructions, process and return results from all different underlying cloud systems. The architecture of sky computing is shown in fig. 5.10.

Each cloud provider, has a specific API that makes available an interaction with their own resources. All these can be aggregated by a middleware layer, which allows controlling and managing resources by translating every command to the correspondent provider API. Abstraction, from bottom to top, is the key for building a consistent system. The upper layer, sky computing, integrates the last level of Infrastructure as a Service and the next layer of Software as a

Service. This is a critical layer, as it must be as comprehensive as possible in features and capabilities. Here, our main focus is HPC, but is must be possible to deal with other applications too. Management, with scheduling, accounting and billing. Should be well developed as well as monitoring and job submission.

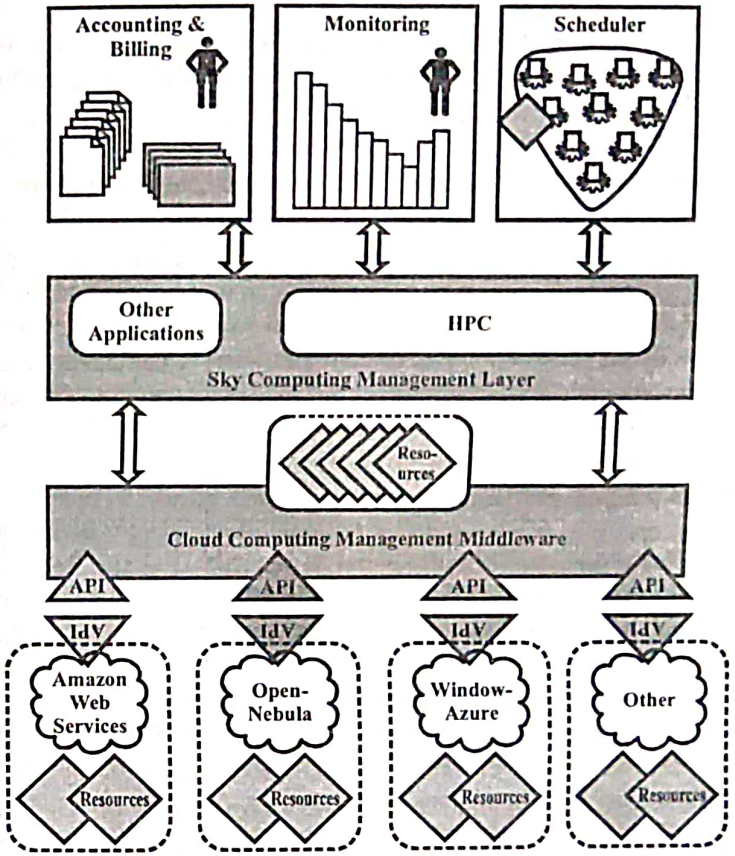


Fig. 5.10 Sky Computing Architecture

(i) Accounting and Billing – When providing users with a complex infrastructure like sky computing, it is crucial that the right usage is being kept for accounting and billing. Assuring a righteous accounting can make monthly usage use prediction, history analysis and the right planning for future use. The saved data also allows to bill the registered users for the used resources, both private and public combined.

(ii) Monitoring Software – Monitoring is also a very important part of cloud management. Probing the resources allows to register and control resource usage for a healthy running. For instance, detecting problems (out of memory, power off, overheat CPU, etc.) prematurely for an early resolution.

Nagios is a monitoring system that enables organizations to identify and resolve IT infrastructure problems before they affect critical business processes. It delivers awareness of IT infrastructure's status and allows to detect and repair problems and mitigate future issues before they affect users.

(iii) Customizable Scheduler – A scheduler is a running daemon that coordinates the virtual requests and the available resources using different scheduling policies. It basically assigns to each virtual machine (VM) a physical host and a storage area depending on resource availability, obeying to pre-defined policies. Neither Deltacloud nor Aeolus have a scheduler, they make the deployment and rely on the destination cloud's management. Some open-source projects available are Haizea and cloud scheduler.

(iv) Cloud Computing Middleware – Middleware is a very important and useful part in the chain value. It provides an abstraction that allows to develop applications without being tied to an explicit cloud vendor. The drawback is that API operations are limited (providers' operation set is larger) and can correspond to loss of performance. The sky computing management layer relies on the lower layer resources and interface, so it should be extremely stable and dependable. There are some projects undergoing for middleware, like the open-source libcloud, Deltacloud, jclouds, or fog, while others, like abiquous, Kaavo or Enstratius offer a more professional customized service and support, in exchange for a monthly fee.

(v) System Assembling – The hardest part is to connect all pieces of the puzzle, thus is was successful. We managed to get Aeolus working with a hybrid infrastructure, featuring Amazon and OpenNebula with a custom scheduler Haizea and Ganglia. The structure was functional and stable, however the lack of some important pieces reduced the structure flexibility and agility, despite the occasional improvement by a new tweaks on fresh software updates.

Q.33. Discuss the characteristics of sky computing.

Ans. The characteristics of sky computing are as follows –

(i) Flexibility and Scalability – The sky can quickly scale upto thousands of servers or services to make resources available as they are needed. Most cloud providers are extremely reliable in providing their services, with many maintaining 99.99% uptime. The connection is always on and as long as workers have an internet connection, they can get to the applications they need from practically anywhere. Some applications even work off-line.

(ii) Security and Trust – In the past, site owners could not trust a remote resource because they had no control over its configuration.

Now that clouds let users control remote resources, however, this concern is no longer an issue. Combining the ability to trust remote sites with a trusted networking environment, a virtual site can now exist over distributed resources.

(iii) Efficiency – Advances in processing, communication and systems/middleware technologies had as a result new paradigms and platforms for computing.

(iv) Flexible Costs – The costs of sky computing are much more flexible than traditional methods. Companies only need to commission and thus only pay for server and infrastructure capacity as and when it is needed. More capacity can be provisioned for peak times and then de-provisioned when no longer needed. Traditional computing requires buying capacity sufficient for peak times and allowing it to sit idle the rest of the time.

(v) Resource Management – Sky computing facilitates the implementation and realization of emerging technologies to deliver better customer experience with improved and real-time interaction across the business operations to maximize the value for the consumer and stakeholders where sustainability can be achieved with increased profitability and competitiveness.

Q.34. Describe the various sky computing provider.

Ans. The various sky computing provider are as follows –

(i) Appliance Providers – Appliances can integrate the information using any configuration method from any appliance provider. This information in the templates is application specific and potentially different from appliance-to-appliance, but the templates themselves are uniform, and any context broker can process them. Example – Amazon was the first major could provider, Amazon Simple Storage Service (Amazon S3), Apple, Cisco, Citrix, IBM, Joyent, Google, Microsoft, Rackspace and Salesforce.

(ii) Cloud Broker – An entity that manages the use, performance and delivery of cloud services and intermediates the relationships between cloud providers and cloud consumers and negotiation, configuration done manually. Example – AWS marketplace from Amazon, BlueWolf, CloudCompare, CloudMore, which offers cloud services aggregation and activation through partners. The company serves the UK, Sweden, Finland, Denmark, Ireland, and more. Key partners include IBM, Microsoft, HP Autonomy, VMWare, and Cryptozone.

(iii) SaaS (Software as a Service) – It represent the largest cloud market and are still growing quickly. SaaS uses the web to deliver applications that are managed by a third-party vendor and whose interface is accessed on the clients' side. Examples – Google Apps, Salesforce, Workday, Concur, Citrix Go to Meeting, Cisco Web ExCommon.

(iv) PaaS (Platform as a Service) – These are used for applications, and other development, while providing cloud components to software. PaaS makes the development, testing, and deployment of applications quick, simple, and cost-effective. With this technology, enterprise operations, or a third-

party provider, can manage OSes, servers, storage, networking, and the PaaS software itself. Examples – EngineYard, RedHat OpenShift, Google App Engine, Heroku, appFog (aF), Windows Azure, Amazon Weg Service(AWS).

(v) IaaS (Infrastructure as a Service) – These are self-service models for accessing, monitoring, and managing remote data center infrastructures, such as compute (virtualized or bare metal), storage, networking, and networking services (e.g. firewalls). Instead of having to purchase hardware outright, users can purchase IaaS based on consumption. Example – Amazon Web Services (AWS), Cisco Metapod, Microsoft Azure, Google Compute Engine (GCE).

Q.35. What is load balancing ? What are the advantages of load balancing ? [R.G.P.V., May 2019 (BE)]

Ans. Load balancing is used for distributing a larger processing load to smaller processing nodes for enhancing the overall performance of system. In cloud computing environment load balancing is required distribute the dynamic local workload evenly between all the nodes.

Load balancing helps in fair allocation of computing resource to achieve a high. User satisfaction and proper resource utilization. High resource utilization and proper load balancing helps in minimizing resource consumption. It helps in implementing fail over, scalability, and avoiding bottlenecks. Load balancing is a techniques that helped networks and resources by providing a maximum throughput with minimum response time. Load balancing is dividing the traffic between all servers, so data can be sent and received without any delay with load balancing. In cloud environment many algorithms are available that helps in proper traffic loaded between all available servers. Most of them can be applied in the cloud environment with suitable verifications. In cloud computing environment, load balancing algorithms can be divided into two main groups – first algorithm type is batch mode heuristic scheduling algorithms (BMHA) and second is online mode heuristic algorithms. In BMHA, jobs are combined together when they are arriving in the system. The BMHA scheduling algorithm will start after a fixed time period. The examples of BMHA based algorithms are – First come first served scheduling algorithm (FCFS), round robin scheduling algorithm (RR), min min algorithm and max min algorithm. In on-line mode heuristic scheduling algorithm, all jobs are scheduled when they are arriving in the system. The cloud environment is a heterogeneous system and in this speed of each processor varies quickly and easily. The online mode heuristic scheduling algorithms are more appropriate and better for a cloud environment. It is very important to estimate proper load, need to do comparison of all load, stability of all different systems, performance of purposed system, interaction between all the nodes and nature of work to be transferred while developing a load balancing algorithm. The most important thing is selecting

the nodes and its also include many other ones. CPU load, amount of memory required combine together to calculate the load of machine. In our daily life example of load balancing is websites. Users could experience many problems without load balancing like delays, timeouts and long system responses.

Fig. 5.11 shows different load balancing algorithms. This is mainly divided into two categories – static load balancing algorithm and dynamic load balancing algorithm –

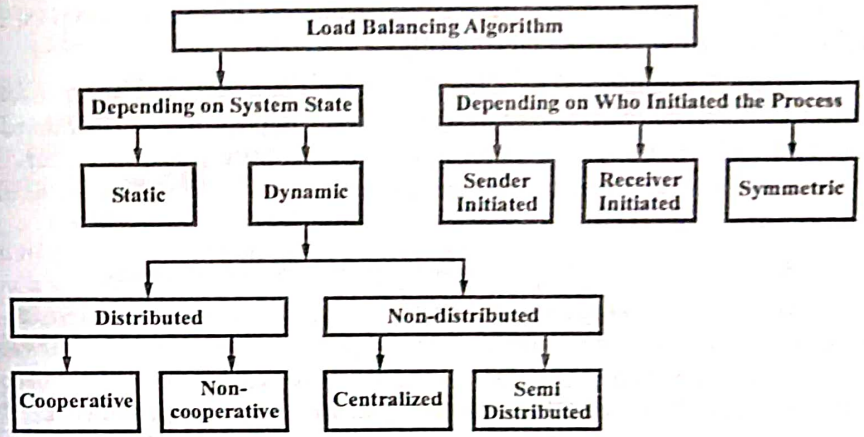


Fig. 5.11 Types of Load Balancing Algorithm

(i) Static Approach – This approach is mainly defined in the design or implementation of the system. Static load balancing algorithm divide the traffic equivalently between all servers.

(ii) Dynamic Approach – This approach considered only the current state of the system during load balancing decisions. Dynamic approach is more suitable for widely distributed systems such as cloud computing.

Dynamic load balancing approaches have two types. They are distributed approach and non-distributed (centralized) approach. It is defined as following–

(a) Centralized Approach – In centralized approach, only a single node is responsible for managing and distribution within the whole system. Other all nodes are not responsible for this.

(b) Distributed Approach – In distributed approach, each node independently builds its own load vector. Vector collecting the load information of other nodes. All decisions are made locally using local load vectors. Distributed approach is more suitable for widely distributed systems such as cloud computing.

Advantages of Load Balancing –

(i) Scalability – The main advantages of the load balancing algorithm is that any number of servers can be added easily without causing any

disturbance and application can be performed smoothly through load balancing the servers in the cloud.

(ii) Performance – An efficient load balancing helps to provide the cloud services and cloud applications to respond faster compare to the usual completion time. Moreover the execution time also get reduced to the greater extent through efficient compression techniques, and caching mechanism.

(iii) Availability – Load balancing mechanism guarantees to provide the services efficiently. In the case of unavailability of the few servers, the load will be further distributed efficiently.

(iv) Reliability – The reliability of the cloud services are protected by the redundancy of the server through which an application can be hosted at any cloud hub in the world. Even in case of the failure the cloud serving resource will not stop functioning and the services will be redirected to any other cloud location.

Q.36. Write short note on resource optimization techniques.

Ans. In the cloud IaaS (Infrastructure as a Service), the equipment is provided in the form of virtual machines running by a hypervisor software. Each virtual machine is characterized by a set of hardware resources, consisting essentially of CPU, memory and external storage network. The provisioning of virtual machines is on-demand and dynamically allocated to users.

The general architecture of cloud shows the important place that takes virtualization software in the overall structure of the system. So any optimization at this level reflects positively on the overall performance of the cloud. Several resource optimization techniques are available, among these, especially there is live migration, load balancing and dynamic reconfiguration.

(i) Live Migration – This process allows moving virtual machines from one physical node to another without service interruption and with a completely transparent way for the user.

(ii) Load Balancing – It can generally allocate workloads services while reducing the number of servers and improving performance.

(iii) Dynamic Reconfiguration – Dynamic reconfiguration of virtual machines provide the ability to modify the CPU power, the size of the memory associated with a virtual machine (VM) without stopping its execution.

It appears as a new attractive solution particularly for resource optimization techniques. In cloud system, the client order resources in the form of a lease, but in general they uses less resources than requested. This is a loss for the client because of unused resources and also can be a loss for the provider.

Q.37. Explain the architecture of cloud with dynamic reconfigurable component.

Ans. Cloud client typically lease virtual machines that include a fixed amount of resources, such as the number of cores, memory size, and so on. These resources are generally stable throughout the life of the virtual machines. During use of the cloud, the resources used may be upper or lower to the resources leased by the client. It can cause a gap between the needs and provisions. The DRC component comes into play to solve this problem.

The implementation of the dynamic reconfigurable component (DRC) is required to optimize the use of cloud computing resources.

The DRC component is intended to correct the users resources requests by removing unused resources, which results in the difference between the resources requested and used.

The DRC component as shown in fig. 5.12 will read the resources used in real time, these resources are measured in accordance with given frequency by key performance indicators (KPI) measuring tool. After, the component will assign new values of resources depending on DRC optimization process and inject them to the cloud manager.

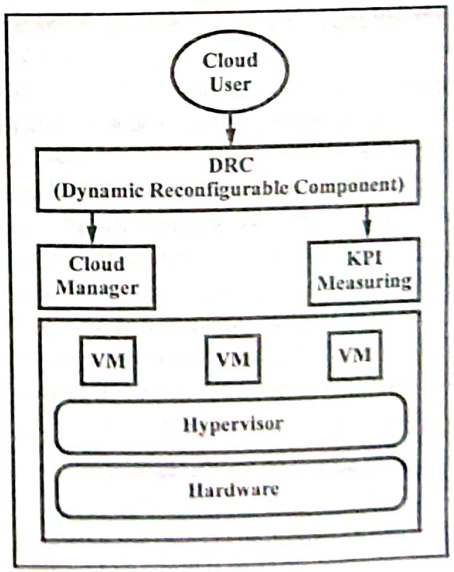


Fig. 5.12 Architecture Cloud with DRC

The KPI measuring will be used to collect measurements data, and will be transferred to the component for apply policy.

The component is intended not to be a part of the cloud manager. It is acting on the outside of cloud manager. The component is designed to be

multiplatforms, it is independent of cloud management tools such as Openstack, Eucalyptus, Cloudstack and OpenNebula. The DRC component is designed for the optimization of material resources and other types of resources. The component could be used for the client as well as for the provider.

MONITORING IN CLOUD, INSTALLING CLOUD PLATFORMS AND PERFORMANCE EVALUATION, FEATURES AND FUNCTIONS OF CLOUD COMPUTING PLATFORMS

Q.38. Discuss the cloud monitoring.

Ans. In clouds, monitoring is essential for the health of the system and is important for both providers and consumers. Primarily, monitoring is a key tool firstly for managing software and hardware resources and secondly for providing continuous information for those resources as well as for consumers' hosted applications on the cloud. Cloud activities like resource planning, resource management, data center management, SLA management, billing, troubleshooting, performance management, and security management essentially need monitoring to effective and smooth operations of the system. Consequently, there is a strong need for monitoring looking at the elastic nature of cloud computing.

In cloud computing, monitoring can be of two types – high-level and low-level. High-level monitoring is related to the virtual platform status. The low-level monitoring is related to information collected for the status of the physical infrastructure. Cloud monitoring system is a self-adjusting and typically multi-threaded system that is able to support monitoring functionalities. It comprehensively monitors pre-identified instances/resources on the cloud for abnormalities. On detecting an abnormal behaviour, the monitoring system attempts to auto-repair this instance/resource if the corresponding monitor has a tagged auto-heal action. In case of auto-repair failure or an absence of an auto-heal action, a support team is notified. Technically, notifications can be sent by different means such as e-mail, or SMS.

Q.39. Write short note on performance management in cloud monitoring.

Ans. Being the hardware infrastructure maintenance delegated to the providers, the cloud computing model is attractive for most consumers (primarily medium sized enterprises and research groups). However, despite the attention paid by providers, some cloud nodes may attain performance orders of magnitude worse than other nodes. If a consumer adopts a public cloud to host a mission-critical service or for a scientific application, performance variability and availability become a concern. Therefore, from a consumer's perspective, monitoring the perceived performance is necessary

to adapt to the changes or to apply corrective measures. For instance, a consumer may decide to host applications at multiple clouds to ensure high-availability, switching between clouds depending on the measured performance. Monitoring is then necessary since it may considerably improve the performance of real applications and affect activity planning and repeatability of experiments.

Q.40. Discuss the following commercial monitoring tools –

(i) RevealCloud (ii) Monitis
(iii) LogicMonitor (iv) Nagios
(v) CloudWatch (vi) Nimsoft.

Ans. (i) RevealCloud – CopperEgg provides RevealCloud monitoring tool. It was founded in 2010 and Rackspace is a main partner. RevealCloud enables its consumers to monitor across cloud layers e.g. SaaS, PaaS, and IaaS. It is not dedicated to only one cloud resources provider, rather it is generic to allow a consumer to get its benefits within most popular cloud providers e.g. AWS EC2, Rackspace, etc. RevealCloud is one of the very few monitoring tools that supports maintaining monitored historical data, it can trace upto last 30 days data, which is considered as a prime feature that most commercial monitoring tools lack.

(ii) Monitis – This was founded in 2005, has one unified dashboard where consumers can open multiple widgets for monitoring. A Monitis consumer needs to enter his/her credentials to access the hosting cloud account. In addition, a Monitis consumer can remotely monitor any website for uptime, in-house servers for CPU load, memory, or disk I/O, by installing Monitis agents to retrieve data about the devices. A Monitis agent can also be used to collect data of networked devices in an entire network (behind a firewall), this technique is used instead of installing a Monitis agent on each single device. Widgets can be also emailed as read only version to share the monitored information. Moreover, Monitis provides rich features for reporting the status of instances where consumers can specify the way a report should be viewed e.g. chart, or graph. It also enables its consumers to share the report publicly with others.

(iii) LogicMonitor – It was founded in 2008 and it is a partner with several third parties such as NetApp, VMWare, Dell, and HP. Similarly to RevealCloud, LogicMonitor enables its consumers to monitor across cloud layers e.g. SaaS, PaaS, and IaaS. It also enables them to operate monitoring operations on multi-cloud resources. Protocol used in communications is SSL outgoing only encrypted connections. Moreover, LogicMonitor uses simple network management protocol (SNMP) as a method of retrieving data about distributed virtual and physical resources.

(iv) Nagios – It was founded in 2007, Nagios supports multi-layer monitoring. It enables its consumers to monitor their resources on different

cloud infrastructure as well as in-house infrastructure. Nagios utilizes SNMP for monitoring networked resources. Moreover, Nagios has been extended with monitoring functionalities for both virtual instances and storage services using a plugin-based architecture. Typically, a Nagios server is required to collect the monitoring data, which would place it as a centralized solution. Moreover, Nagios is a cloud solution as a user would need to setup a Nagios server. However, many possible configurations can help create multiple hierarchical Nagios servers to reduce the disadvantages of a centralized server.

(v) CloudWatch – It is one of the most popular commercial tools for monitoring the cloud. It is provided by Amazon to enable its consumers monitoring their resources residing on EC2. Hence, it does not support multi-cloud infrastructure monitoring. The technical approaches used in CloudWatch to collect data are implicit and not exposed to users. CloudWatch is limited in monitoring resources across cloud layers. However, an API is provided for users to collect metrics at any cloud layer but requires the users to write additional code.

(vi) Nimsoft – It was founded in 2011. Nimsoft supports multi-layers monitoring and both virtual and physical cloud resources. Moreover, Nimsoft enables its consumers to view and monitor their resources in case they are hosted on different cloud infrastructures e.g. a Nimsoft consumer can view resources on Google Apps, Rackspace, Amazon, salesforce.com and others through a unified monitoring dashboard. Also, Nimsoft give its consumers the ability to monitor on both private and public clouds.

Q.41. Explain in detail about the OpenStack.

Ans. OpenStack is a set of software tools for building and managing cloud computing platforms for public and private clouds as well as it is a collection of open source software projects which provides an Infrastructure-as-a-Service (IaaS) solution through a set of interrelated services. In July 2010 NASA and Rackspace Hosting combindly launched an open-source cloud-software initiative which is known as OpenStack. OpenStack code came from NASA's Nebula platform and from Rackspace's Cloud Files platform. OpenStack, project is an open source cloud computing platform which provide the ubiquitous open source cloud computing platform for public as well as private clouds, also it is free and open-source software released under the terms of the Apache License. The cloud is mainly providing computing features for end users in a remote environment, where the actual software runs as a service on reliable, scalable servers rather than on each end users computer. OpenStack give facilities for deploying virtual machines (VMs) and other instances which handle different tasks for managing a cloud environment. It provides horizontal scaling very easy, which means that tasks which benefit

from running concurrently can easily serve more as well as less users on the fly by just spinning up more instances.

Components of OpenStack are as follows –

(i) Object Storage (Swift) – OpenStack Object Storage (Swift) is a scalable redundant storage system for objects and files. Objects as well as files are written to multiple disk drives spread throughout servers in the data center, OpenStack software only responsible for ensuring data replication and integrity across the cluster.

(ii) Compute (Nova) – OpenStack Compute (Nova) is a cloud computing fabric controller, which is used for deploying and managing large numbers of virtual machines and other instances to handle computing tasks.

(iii) Block Storage (Cinder) – OpenStack Block Storage (Cinder) is a block storage component, which is more analogous to the traditional notion of a computer being able to access specific locations on a disk drive as well as it provides persistent block-level storage devices for use with OpenStack compute instances. In OpenStack, the block storage manages the creation, attaching, detaching of the block devices to servers.

(iv) Database (Trove) – OpenStack (Trove) is a database as a service which provides relational and nonrelational database engines.

(v) Networking (Neutron) – OpenStack Networking (Neutron) provides the networking capability for OpenStack and it is a system for managing networks and IP addresses easily, quickly and efficiently.

(vi) Identity Service (Keystone) – OpenStack Identity (Keystone) provides identity services for OpenStack or it is a central directory of users mapped to the OpenStack services they can access. It provides multiple means of access, and acts as a common authentication system across the cloud operating system and can integrate with existing backend directory services like LDAP.

(vii) Dashboard (Horizon) – OpenStack Dashboard (Horizon) is the dashboard behind OpenStack which provides administrators and users a graphical interface to access, provision and automate cloud-based resources.

(viii) Image Service (Glance) – OpenStack Image Service (Glance) provides image services to OpenStack, discovery, registration and delivery services for disk and server images, it also allows these images to be used as templates when deploying new virtual machine instances.

(ix) Orchestration (Heat) – OpenStack Orchestration (Heat) is a service which allows developers to store the requirements of a cloud application in a file that defines what resources are necessary for that application.

(x) Telemetry (Ceilometer) – OpenStack Telemetry Service (Ceilometer) provides telemetry services, which allows the cloud to provide

billing services to individual users of the cloud, it keeps a verifiable count of each user's system usage of each of the various components of an OpenStack cloud.

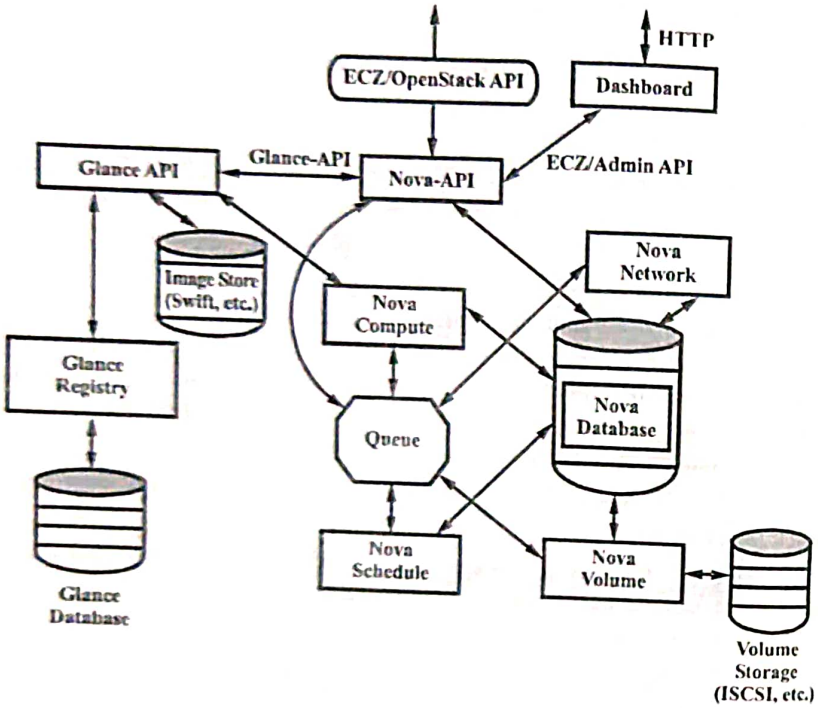


Fig. 5.13 OpenStack Architecture

Q.42. How to install OpenStack on RHEL 6.4.

Ans. **Installation Process of OpenStack on RHEL 6.4 –**

First of all install RHEL 6.4 on your Personal Computer (PC), then follow these commands on terminal to install OpenStack –

(i) yum install kernel*openstack*
reboot

(ii) check or verify kernel is openstack or not
uname –r

(iii) yum install openstack-packstack

(iv) packstack --gen-answer-file=lwx.text

(v) vim lwx.txt
Change this information –

CONFIG_NTP_SERVERS=ip address of the main server (for example: 192.168.0.254) -line number 33 of the configuration file.
CONFIG_HORIZON_SSL=y -line number 225 of the configuration file.

(vi) packstack --answer-file=lwx.text

(vii) ovs-vsctl add-port br-ex eth0
dhclient –v br-ex

(viii) cd/etc/sysconfig/network-scripts/
mv ifcfg-br0 ifcfg-br-ex
vim ifcfg-br-ex
Chage this information –
//Bridge Networking Interface

DEVICE=br-ex	-write in configuration file.
ONBOOT=yes	-write in configuration file.
USERCTL=yes	-write in configuration file.
NM_CONTROLLED=no	-write in configuration file.
IPV6INIT=no	-write in configuration file.
PEERNTP=no	-write in configuration file.
DELAY=0	-write in configuration file.
IPADDR=192.168.0.10	-write in configuration file.
GATEWAY=192.168.0.254	-write in configuration file.
DNS1=192.168.0.254	-write in configuration file.

(ix) vim ifcfg-eth0
Change this information –

DEVICE=eth0	-write in configuration file.
ONBOOT=yes	-write in configuration file.

(x) service network restart

(xi) reboot

Q.43. List out the performance evaluation criteria of cloud computing.

Ans. Some important criteria for performance evaluation of cloud computing are as follows –

(i) Average response time per unit time, this criterion will cover all factors completely.

(ii) Network capacity per second (Mbps) or unit time, the most important factor associated with this criterion is network bandwidth, availability and scalability.

(iii) The number of I/O commands per second (IOPS) or unit time.

(iv) Workload (requests) to be serviced per second (Mbps) or a unit of time.

(v) Average waiting time per unit time.

(vi) Throughput (Req/Sec), this criterion will cover recovery, buffering capacity and processing power factors.

(vii) The average time of processing (exe/sec).

(viii) Percentage of CPU utilization.

(ix) The number of requests executed per unit time.

(x) The number of requests per unit time buffer.

(xi) The number of rejected requests per unit time.

Q.44. Explain in detail about Google Cloud platform.

Ans. Developers can code, test and deploy their applications with highly scalable and reliable infrastructure that is provided by Google and Google itself uses it. Developers have to just pay attention to the code and Google handles issues regarding infrastructure, computing power and data storage facility. Google Cloud is one of the popular IoT platform because of – Fast global network, Google's BigData tool, Pay as you use strategy, Support of various available services of cloud like RiptideIO, BigQuery, Firebase, PubSub, Telit Wireless solutions, Connecting Arduino and Firebase and Cassandra on Google cloud platform and many more.

Fig. 5.14 shows real time stream processing by Google. Devices send their status information to App Engine. So first load balancer makes sure that the load is balanced among various app engines. Then compute engine performs data computation and publication. Multiple instances of compute engine are available to insure reliability and scalability. The data is stored and backed up using cloud storage. Big query allows speedy insertion of data in tables of cloud database. The results can be presented to the end users by means of various analysis and visualisation technique.

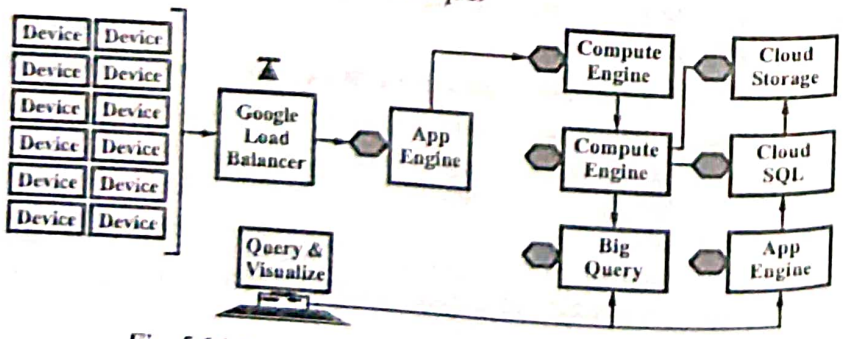


Fig. 5.14 Real Time Stream Processing Google IoT

Q.45. Explain various characteristics of Google Cloud platform.

Ans. Important characteristics of Google Cloud platform are as follows –

(i) Streaming Insights – Events of interest fire off continuously in the physical world, and data that is required for decision making cannot always wait for offline analysis. Internet-equipped sensors on any physical item imaginable make it possible to ingest data continuously into the cloud, directly from the source at massive scale.

(ii) Tap into the World – A new type of device technology along with ubiquitous networking makes it easy and economical to mine information from any physical item and place. This untapped pool of data gives organizations visibility into parts of their operations previously considered "offline". Combined with real-time processing and predictive analytics, an IoT capability profoundly changes monitoring and management practices by enabling proactive resolution in response to real-time events, and ultimately, predictive capabilities.

(iii) From Small to Big (Data) – Each sensor-equipped device may be small and yields only incremental insight. Multiply this by hundreds, thousands, or millions of sensors all ingesting data to the cloud and the collective stream presents as a big data problem. Cloud Pub/Sub makes real-time, reliable processing of IoT data easy, and cloud storage products persist all big data efficiently and economically. IoT on cloud platform allows us to make extremely fast queries into any business and operating environment, without managing any infrastructure.

(iv) Global Fiber Network – Google operates its own private fiber network that spans the globe with over 70 points of presence across 33 countries, ensuring data to and from your devices gets delivered at ultra-low latency. Reliability and security are enhanced because IoT data do not have to travel the public Internet through the majority of its time in transit. Google's global network ensures that millions of devices and sensors distributed worldwide can deliver raw data efficiently so an organization can tap operational insight continuously with no disruption.

(v) Google-Grade Security – Whether device-to-cloud or cloud-to-device, security is the most important concern as IoT is increasingly used to support business-critical operations. All cloud platform APIs are secure by default with full encryption, backed by integrated and pervasive security across the entire infrastructure. Cloud IAM can ensure devices have access only to resources we explicitly designate.

Q.46. Write short note on cloud computing platform.

Ans. Open cloud platforms provide flexibility, on demand services and allow great amount of customization. The open source cloud platform provides features to end-user for improved scalability, portability, and flexibility as well

as on-demand basis services. The most popular and commonly used open source software such as CloudStack, Eucalyptus, Nimbus, OpenStack, OpenNebula, Xen cloud platform, TPlatform, Apache virtual computing lab (VCL) and Enomaly elastic computing platform. The analysis and summarization would help the users to understand the characteristics and would allow users to choose better services according to their requirements and also make more unified decision on the open source cloud platform according to their compatibility, scalability, implementation, interfaces, deployment requirement, and development support. Since cloud computing is a most important and evolving technology there are many features which are being added the comparison is based on the current features as well as technology available in these all open source platform however there is need for incorporation, enterprises of more features to improve these framework.

Q.47. Explain in detail about the Xen cloud platform.

Ans. The Xen cloud platform (XCP) manages storage, VMs and the network in a cloud. XCP does not provide the overall cloud architecture, but rather focuses on configuration and maintenance of clouds. It also enables external tools, including Eucalyptus and OpenNebula, to better leverage the Xen hypervisor.

Fig. 5.15 shows the XCP architectural components. Its basic component is the XCP host, which is a Xen hypervisor enabled to communicate with other XCP hosts. Several XCP hosts can be bound together into a XCP resource pool. A single XCP host from this pool must be setup as the master XCP host, which offers an administration interface and commands other XCP hosts. Optionally, a resource pool may have a shared storage whose objective is to store and export VM images mainly for VM migration, which allows administrators to place and replace VMs on any XCP host.

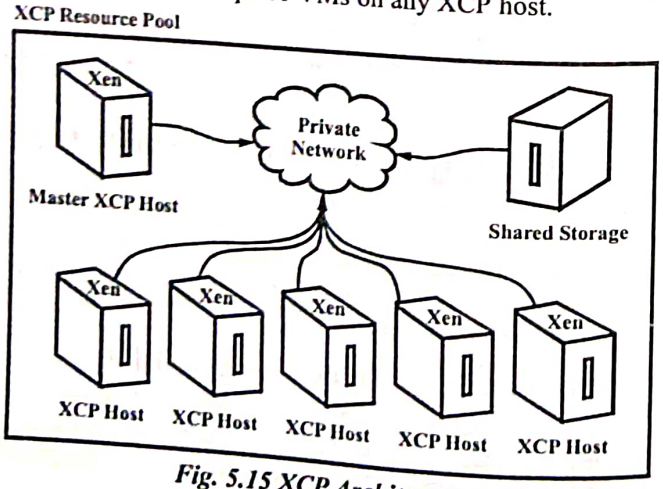


Fig. 5.15 XCP Architecture

XCP networking deserves dedicated attention by itself. It is based on the open switch project. The approach distinguishes virtual from physical interfaces while offering software based VLAN support. The open switch provides three software components – the physical network interface (PIF), the virtual network interface (VIF), and the virtual ethernet switch (VES). PIFs represent physical interfaces attached on a XCP host. Similarly, VIFs represent interfaces attached on the VM. The VES is a virtual switch on a XCP host, which can be used to connect VIFs with each other and with the PIF. A VES without an association to a PIF can be used to provide connectivity only between VMs on a given XCP host, with no connection to the outside world as shown in fig. 5.16.

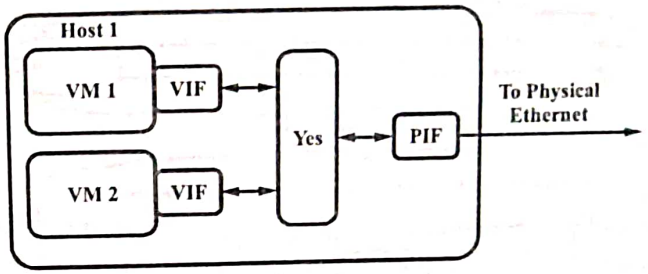


Fig. 5.16 XCP Network Architecture

XCP supports VLANs through the use of additional PIFs corresponding to specific VLAN tags. Thus, it is possible to see all traffic on the physical network interface using a VES attached to a PIF associated with the interface.

Inter-host Communication – XCP provides a management infrastructure with an appropriate API to install, monitor and manage various aspects of the VM infrastructure. The API calls make use of the XML-RPC protocol to transmit requests and responses over the network. These XML-RPC requests and responses may also be exchanged between hosts in a XCP resource pool through HTTP protocol. If desirable, this inter-host communication can be turned secure using SSL-encrypted HTTP (HTTPS).

Q.48. Explain conceptual representation of eucalyptus cloud. Also explain in brief the components within eucalyptus system.
[R.G.P.V., Dec. 2013 (BE)]

Ans. Eucalyptus (Elastic Utility Computing Architecture Linking Your Programs to Useful Systems) is an open source software infrastructure for implementing on-premise clouds built on top of existing IT and service provider infrastructure. Thus, with a eucalyptus private cloud, sensitive data remains secure from external intrusion behind the enterprise firewall. Eucalyptus was designed to ensure its compatibility with existing Linux-based data center installations, and can be deployed without modification on all major Linux OS distributions. The software framework is a highly modular cooperative set of

Web services that interoperate based on standard communication protocols. Through, this framework, virtual machines and storage resources are interconnected on an isolated layer 2 network.

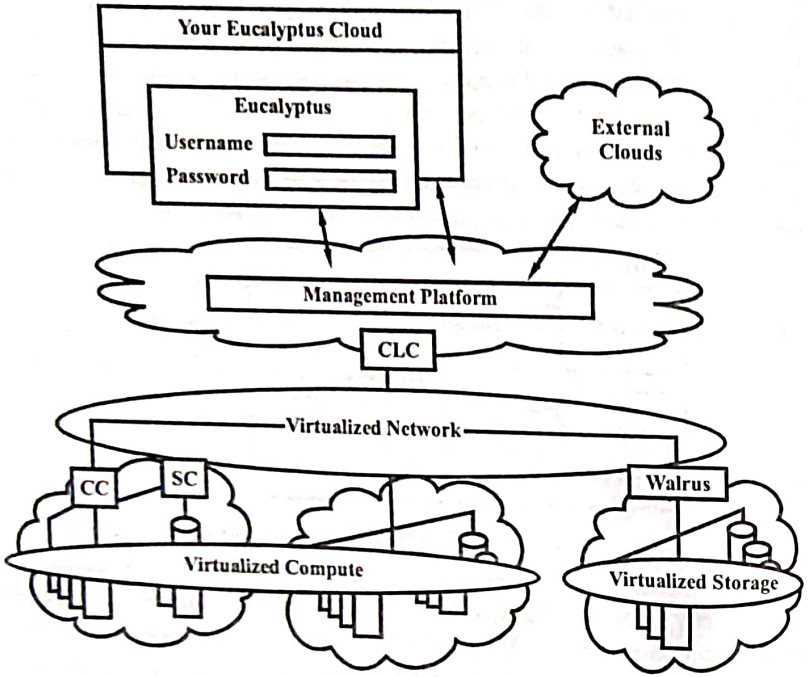


Fig. 5.17

Fig. 5.17 shows a conceptual representation of eucalyptus cloud system. Each eucalyptus service component exposes a well defined language agnostic API in the form of a WSDL document containing both the operations that the service can perform and the input/output data structures. The components within the eucalyptus system are as follows –

(i) Cloud Controller (CLC) – It is the entry point into the cloud for administrators, developers, project, managers, and end-users. The CLC is responsible for querying the node managers for information about resources, making high level scheduling decisions, and implementing them by making requests to cluster controllers. The CLC is also the interface to the management platform. In essence, the CLC is responsible for exposing and managing the underlying virtualized resources via a well defined industry standard API and a Web-based user interface.

(ii) Cluster Controller (CC) – It executes on a cluster front-end machine, or any machine that has network connectivity to both the nodes running node controllers and to the machine running the CLC. CCs gather

information about a set of VMs and schedules VM execution on specific node controllers. The CC also manages the virtual instance network and participates in the enforcement of SLAs as directed by the CLC. All nodes served by a single CC must be in the same broadcast domain.

(iii) Node Controller (NC) – It is executed on every host that is designated for hosting VM instances. NCs control the execution, inspection, and termination of VM instances on the host where it runs, fetches and cleans up local copies of instance images, and queries and controls the system software on its node in response to queries and control request from the CC. The NC is also responsible for the management of the virtual network endpoint.

(iv) Storage Controller (SC) – It implements block accessed network storage and is capable of interfacing with various storage systems. An elastic block store is a Linux block device that can be attached to a virtual machine but sends disk traffic across the locally attached network to a remote storage location. An EBS volume cannot be shared across instances but does allow a snap-shot to be created and stored in a central storage system.

(v) Walrus – It allows users to store persistent data, organized as eventually-consistent buckets and objects. It allows users to create, delete, list buckets, put, get, delete objects and set access control policies. Walrus is interface compatible with Amazon's S3, and supports the Amazon machine image image-management interface, thus providing a mechanism for storing and accessing both the virtual machine images and user data.

(vi) Management Platform – It provides an interface to various eucalyptus services and modules. These features can include VM management, storage management, user/group management, accounting, monitoring, SLA definition and enforcement, cloud-bursting, provisioning, etc.

Q.49. Explain in detail about the OpenNebula.

Ans. OpenNebula is a flexible tool that orchestrates storage, network and virtualization technologies to enable the dynamic placement of services on distributed infrastructures. A number of communities are actively using OpenNebula. Some of these are – the European Space Astronomy Centre and the European Organization for Nuclear Research (CERN).

OpenNebula has been designed to be modular in order to allow its integration with as many different hypervisors and environments as possible. It assumes that the physical infrastructure adopts a classical cluster-like architecture with a front-end, and a set of host nodes where VMs will execute. There is at least one physical network joining all the cluster nodes with the front-end. The front-end executes the main OpenNebula processes while the cluster nodes are hypervisor-enabled hosts that provide the resources needed by the VMs.

OpenNebula is designed with three layers in mind – tools, core and drivers, as depicted in fig. 5.18. The tools layer contains modules providing functionalities for administrators and clients. One component is the command line interface (CLI) that can be used by administrators to manipulate the infrastructure through intuitive commands. The scheduler module, responsible for VM placement, is implemented in this layer. Other tools can be created using he OpenNebula cloud API which is based on a XML-RPC interface.

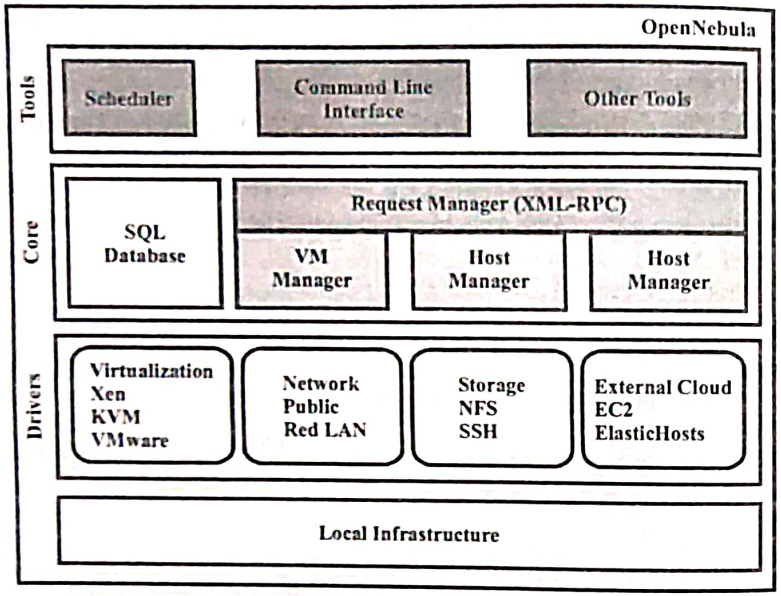


Fig. 5.18. Components of OpenNebula

Similarly to Eucalyptus, OpenNebula works with administrative and client accounts. Administrators access OpenNebula through CLI, while clients launch and manage VMs using Web services interfaces. OpenNebula implements an interface compatible with the EC2 query API from Amazon and another one compatible with the open cloud computing interface from the open grid Forum.

The core layer consists of components responsible for handling client requests and control resources. The main component of this layer is the request manager, which handles client requests through an XML-RPC interface calling internal components according to the invoked method. Hosts and VMs are managed and monitored by the host manager and the VM manager, respectively. The virtual network manager (VN manager) manages virtual networks by keeping track of IP and MAC addresses and their association with VMs. The SQL database stores internal data structures.

Finally, the third layer is formed by modules called drivers that supports different underlying platforms. These drivers run on separated processes

that communicate with the core module through a simple text messaging protocol. There are drivers to deal with file transfers that are implemented by network protocols like NFS and SSH. Also, there are drivers to manage VMs that are dependent on each hypervisor running on the host. Finally, there are drivers to request services from external clouds like Amazon EC2 or ElasticHosts.

OpenNebula manages IP and MAC addresses of VMs and the virtual networks between them. There are two types of virtual networks – the fixed network (public) that uses a fixed set of IP and associated MAC addresses and the ranged network (Red LAN) defined over a range of network addresses. VMs must pertain to one Red LAN and can, optionally, pertain to the fixed network.

Q.50. Discuss the term Nimbus.

Ans. Nimbus is an open source solution (licensed under the terms of the Apache License) to turn clusters into an Infrastructure as a Service (IaaS) for cloud computing focusing mainly on scientific applications. This solution gives to users the possibility to allocate and configure remote resources by deploying VMs – known as virtual workspace service (VWS). A VWS is a VM manager that different front-ends can invoke.

To deploy applications, Nimbus offers a "cloudkit" configuration that consists of a manager service hosting and an image repository. The workspace components are shown in fig. 5.19.

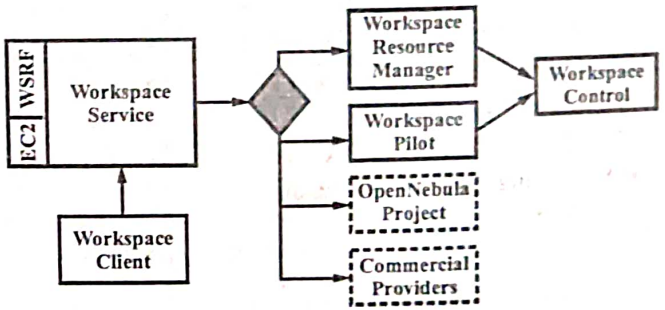


Fig. 5.19 Nimbus Workspace Components

(i) Workspace Service – It is Web services based and provides security with the GSI authentication and authorization. Currently, Nimbus supports two front-ends – Amazon EC2 and WSRF.

(ii) Workspace Control – It is responsible for controlling VM instances, managing and reconstructing images, integrating a VM to the network and assigning IP and MAC addresses. The workspace control tools operate with the Xen hypervisor and can also operate with KVM.

(iii) Workspace Resource Management – It is an open source solution to manage different VMs, but can be replaced by other technologies such as OpenNebula.

(iv) Workspace Pilot – It is responsible for providing virtualization with few changes in cluster operation. This component handles signals and has administration tools.

Q.51. Give comparison of cloud computing platforms.

Ans. The comparison of cloud computing platforms are shown in table 5.1 –

Table 5.1

Features	***Eucalyptus***	***Nimbus***	***OpenNebula***
Cloud Character	Public	Public	Private
Scalability	Scalable	Scalable	Dynamic, scalable
Clouds form	IaaS	IaaS	IaaS
Compatibility	Support EC2, S3	Support EC2	Open, multi-platform
Deployment	Dynamical deployment	Dynamical deployment	Dynamical deployment
Deployment Manner	Command line	Command line	Command line
Transplant-ability	Common	Common	Common
VM support	Xen, VMware, KVM	Xen	Xen, VMware
Web interface	Web service	EC2, WSDL, WSRF	libvirt, OCCI, EC2, API
Structure	Module	Lightweight components	Module
Reliability	–	–	Rollback host and VM
OS support	Linux	Linux	Linux
Development Language	Java	Java, python	Java

●●●